

NEURAL NETWORKS IN VIBRATION ANALYSIS
OF GAS TURBINES

By

GREGORY ANTHONY HARRISON

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

1997

*

Copyright 1997

by

Gregory Anthony Harrison

Dedicated to my wife, Kathy, and son, Christopher.

ACKNOWLEDGMENTS

The assistance and support of the Lockheed Martin Corporation, particularly the people in the Marine Programs and Automation Systems business area, was greatly appreciated and helped tremendously. Thanks go to Clif Wigginton for his expert help with gas turbine questions, his constant support, and assistance in getting the tools and information needed to perform the research. The people in the Lockheed-Martin Engineering Lab were very helpful in the creation of filters and components for the digitizer system.

Thanks go to the people at the Florida Power Cogeneration Plant at the University of Florida for allowing their LM6000 turbine to be recorded. Helpful assistance was also provided by the turbine vibration experts at General Electric and in the US Navy, particularly Bruce Thompson.

The ongoing confidence and support expressed by Dr. Fred Taylor helped make the effort possible. Dr. Principe provided many comments that helped to ensure the validity of the approach. Dr. Arroyo was instrumental in my introduction to computational intelligence. Dr. Harris was an inspiration for many aspects of neural computing. Dr. Hansen allowed me to access the LM6000 turbine at the University and provided guidance on condition monitoring and application of the technology to actual turbomachinery.

TABLE OF CONTENTS

	<u>page</u>
ACKNOWLEDGMENTS.....	iv
ABSTRACT.....	xiii
LIST OF TABLES.....	vii
LIST OF FIGURES.....	viii
 CHAPTERS	
1 INTRODUCTION.....	1
2 GAS TURBINE VIBRATION SIGNAL FEATURES	6
Gas Turbine Application.....	6
LM2500 Gas Turbine.....	7
Data Collection	8
Vibration Transducer.....	21
Neural Network Features.....	25
Time Domain Features	26
Frequency Domain Features	29
3 NEURAL NETWORK SELECTION.....	37
Supervised Neural Networks.....	41
Unsupervised Neural Networks.....	43
Supervised Neural Network Performance Survey.....	45
Data Acquisition, Feature Extraction and Preprocessing.....	45
Neural Network Comparison	50
4 FUZZY ADAPTIVE RESONANCE THEORY NEURAL NETWORK APPLICATION	62
Engineering Description.....	63
Fuzzy ART Theory.....	76
Fuzzy Set Operators	79
Input Layer Description	80
Preprocessing Layer Description	81
Activity Layer and Category Layer Descriptions	83
Orienting Subsystem	93
Dynamic Operation	95
Use of Adaptive Resonance Theory in Vibration Analysis.....	113
Consistency of the Estimator.....	115
The F Vector.....	130
The V Vector.....	137

Case Vigilance Vector	148
The J Matrix	163
Turbine vibration characteristics: a priori information ..	168
Resonance fields	169
Training and testing with expanded input dimensionality ..	186
Multiple-Spectrum Training Sets	190
LM6000 Turbine Application	194
Testing	197
5 CONDITION-BASED MONITORING SYSTEM	203
6 CONCLUSIONS.....	214
Results	214
Contributions	215
Conclusions	217
Recommendations for Further Research.....	221
APPENDICES	
A DIGITIZING EQUIPMENT.....	224
High-Frequency Sampler.....	225
Low-Frequency Long-Duration Sampler.....	227
B ANTI-ALIASING FILTER.....	230
Analog Filter Section.....	231
Digital Filter Section.....	233
LIST OF REFERENCES.....	234
BIOGRAPHICAL SKETCH.....	238

LIST OF TABLES

<u>Table</u>	<u>page</u>
2-1. Vibration test and analysis guide (VTAG) data.....	34
3-1. Features developed in Matlab program.....	48
3-2. One-of-C coding.....	49
4-1. Variance testing configurations.....	124
4-2. Results of single-point classification-region tests.....	163
4-3. Resonance field component calculations.....	176
4-4. 4-D testing results, 51 a priori classes. Epochs: 4, nodes: 215, time 62 sec.	198
4-5. 5-D testing results, 201 a priori classes. Epochs: 4, nodes: 215, time: 57 sec.	199
4-6. 5-D testing results, 241 a priori classes. Epochs: 4, nodes: 215, time: 54 sec.	200
4-7. 5-D testing results, 241 a priori classes, no noise. Epochs: 4, nodes: 215, time: 55 sec.	200
4-8. 5-D testing results, 869 a priori classes. Epochs: 5, nodes: 236, time: 76 sec.	201
4-9. 5-D testing results, 869 a priori classes, no noise. Epochs: 2, nodes: 228, time: 1 sec.	201
6-1. Supervised neural network classification accuracy on the vibration dataset.	214

LIST OF FIGURES

<u>Figure</u>	<u>page</u>
2-1. LM2500 tape one, forward, gas generator vibration.....	12
2-2. LM2500 tape one, forward, power turbine vibration.....	13
2-3. LM2500 tape one, forward, gas generator speed.....	13
2-4. LM2500 tape one, forward, power turbine speed.....	14
2-5. LM2500 tape two, forward, gas generator.....	15
2-6. LM2500 tape two, forward, power turbine.....	15
2-7. LM2500 tape two, forward, gas generator speed.....	16
2-8. LM2500 tape two, forward, power turbine speed.....	16
2-9. LM2500 tape three, forward, gas generator.....	18
2-10. LM2500 tape three, forward, power turbine.....	19
2-11. LM2500 tape three, forward, gas generator speed.....	19
2-12. LM2500 tape three, forward, power turbine speed.....	20
2-13. LM6000 high-pressure turbine, lower power.....	21
2-14. LM6000 low-pressure turbine, lower power.....	22
2-15. LM6000 high-pressure turbine, higher power.....	22
2-16. LM6000 low-pressure turbine, higher power.....	23
2-17. Turbine once-per-rev frequencies (1X) for power turbine (PT) and gas generator (GG), with harmonics (2X and 3X) and exhaust rumble.	31
2-18. A blade-pass feature and sidebands.....	32
2-19. LM2500 gas turbine plan view.....	33
2-20. LM2500 gas turbine combustion noise.....	35
3-1. Supervised network analysis system.....	47

3-2.	DataSculptor data transformation scheme.....	48
3-3.	Back-Propagation Neural Network.....	53
3-4.	Probabilistic Neural Network.....	55
3-5.	Learning Vector Quantization Network.....	56
3-6.	Radial Basis Function Network.....	58
3-7.	Fuzzy ARTMAP Network.....	59
3-8.	Modular Neural Network.....	60
4-1.	Vector illustration of Fuzzy ART.....	69
4-2.	New vector learned by Fuzzy ART.....	70
4-3.	Vectors not resonating with neuron prototype.....	70
4-4.	Effect of a priori coding on input vector space.....	76
4-5.	Fuzzy ART System Diagram.....	78
4-6.	Fuzzy-min of two vectors.....	80
4-7.	Fuzzy-max of two vectors.....	81
4-8.	Complement-coding of vector A , yielding A^c	83
4-9.	Single-point prototype, one vector learned.....	85
4-10.	Rectangular prototype, two vectors learned.....	86
4-11.	Expanded prototype, third vector learned.....	86
4-12.	Activation response to new vector A	91
4-13.	Activation response to closer vector A	92
4-14.	Activation response to small vector A	93
4-15.	Activation response within prototype.....	94
4-16.	Fuzzy ART neural network interconnections.....	96
4-17.	Activity diagram of Fuzzy ART learning.....	99
4-18.	Activity diagram for Fuzzy ART testing and on-line operation.	105
4-19.	Neural network input signal preprocessing.....	116
4-20.	Partitioning of segments for Welch method processing.....	119

4-21.	Example of simulated turbine signal.....	121
4-22.	Variance of different configurations of FFT ensemble averaging.....	128
4-23.	Notched box-whisker plot of experiment groups.....	129
4-24.	Description of box-whisker plot.....	130
4-25.	Raw digitized vibration signal, gas generator.....	132
4-26.	Raw digitized vibration signal, first 500 samples.....	132
4-27.	Calibrated input signal.....	134
4-28.	Calibrated input signal spectrum.....	134
4-29.	Input spectrum, dB.....	135
4-30.	Raw sine wave, with frequency between FFT components.....	139
4-31.	P301 Flat-top window.....	140
4-32.	Windowed sine wave.....	140
4-33.	FFT of raw sine wave.....	141
4-34.	FFT of flat-top windowed sine wave.....	141
4-35.	FFT of FFT of input signal. The DC level causes the upper frequency information to appear near zero amplitude.	144
4-36.	FFT of FFT of input signal with higher components zeroed. Note small extent of non-zero information compared to the complete 32768-component FFT.	144
4-37.	Input spectrum noise-baseline.....	145
4-38.	Input spectrum with noise-baseline standardized.....	146
4-39.	Standardized input signal.....	148
4-40.	Classification Region Diagram.....	152
4-41.	Neural classification regions, $\bar{\rho}=0.2$	153
4-42.	Neural classification regions, $\bar{\rho}=0.5$	154
4-43.	Neural classification regions, $\bar{\rho}=0.976$	155
4-44.	Neural classification regions, $\bar{\rho}=0.976$, no spectrum.....	156
4-45.	Pseudo-cumulative distribution function of standardized spectrum.	158

4-46.	Classification of components using case vigilance.....	159
4-47.	Classification regions using case vigilance.....	160
4-48.	11 th to 16 th Blade set energy before 5% change.....	161
4-49.	11 th to 16 th Blade set classification regions.....	162
4-50.	11 th to 16 th Blade set with 5% change at standardized frequency of approximately 0.53328.	163
4-51.	Locations in a 2-D J-matrix to separate resonance fields associated with 5% change detection.	165
4-52.	Locations in a 3-D J-matrix to separate resonance fields associated with 5% change detection.	166
4-53.	Resonance field for 1-D template containing 1 point.....	172
4-54.	Multipoint template resonance field.....	173
4-55.	Limiting case: template expands to the size of the detectable change threshold.	173
4-56.	Analysis sections for 2-D resonance fields.....	175
4-57.	Example 2-D Resonance Field.....	177
4-58.	Two vectors near resonance field.....	178
4-59.	Response of network to vector A	178
4-60.	Response of network to vector B	179
4-61.	Resonance field test for single classification region.....	180
4-62.	Resonance fields for high-vigilance turbine data.....	181
4-63.	Component analysis identification.....	182
4-64.	Illustration of problem of extended resonance fields.....	183
4-65.	All resonance fields and classification regions from 4-D analysis, shadowed onto a 2-D plane.	186
4-66.	Resonance fields and classification regions for Feature 0....	188
4-67.	Feature dimension for turbine blade set 11-16, resonance fields and classification regions.	188
4-68.	Close-up of resonance fields and classification regions of Feature 0, showing non-interference to feature at standardized frequency of approximately 0.562.	189

4-69.	2-D shadow of all 4-D resonance fields and classification regions for multiple data sets.	192
4-70.	Resonance fields and classification regions for Feature 0, for multiple data sets.	193
4-71.	Feature dimension for turbine blade set 11-16, resonance fields and classification regions for multiple data sets.	193
4-72.	Close-up of resonance fields and classification regions of Feature 0, showing non-interference to feature at standardized frequency of approximately 0.562 for multiple data sets.	194
4-73.	LM6000 spectrum, calibrated per LM2500 sensor.....	195
4-74.	Fuzzy ART classification of LM6000 standardized spectrum.....	196
4-75.	Fuzzy ART classification regions for LM6000 standardized spectrum.	196
4-76.	Pseudo-CDF of LM6000 spectrum.....	197
4-77.	Spectrum changes detected by 4-D neural net.....	199
4-78.	Spectrum changes detected by 5-D neural net, with 869 a priori classifications and no noise.	202
5-1.	Condition-Based Monitoring System.....	210
A-1.	Digitizing equipment.....	225
A-2.	Digitizer and data recorder system.....	226
B-1.	Anti-aliasing filter SPICE circuit.....	232
B-2.	Anti-aliasing filter implementation.....	232
B-3.	DSP IIR biquad signal flow graph.....	233

Abstract of Dissertation Presented to the Graduate School
Of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

NEURAL NETWORKS IN VIBRATION ANALYSIS
OF GAS TURBINES

By

Gregory A. Harrison

December 1997

Chair: Dr. Fred J. Taylor

Major Department: Electrical and Computer Engineering

Artificial neural network technology was applied to the problem of vibration analysis of a large gas turbine engine. The subject domain of gas turbine vibration was researched and neural network input features describing the turbine operation were developed. Different neural network architectures were researched and applied to the vibration analysis. Both supervised and unsupervised neural networks were created and tested, with an emphasis on the unsupervised Fuzzy ART network.

A Fuzzy ART network capable of analyzing a 32768-point vibration spectrum and detecting 96% of changes introduced throughout the spectrum was developed. The theory of resonance fields and the technique of feature separation using a priori information were developed and utilized in the research. Individual case-vigilance techniques were formalized and used to reduce noise clutter in the processing of the spectrum for neural net input. Through training, the

neural network automatically created vibration amplitude acceptance envelopes, or classification regions, through neural compression of the spectrum, to perform continuous monitoring of changes in individual turbine-component vibration. This neural network has application to detection of narrowband vibration trends, important for condition-based maintenance of turbomachinery.

In the course of this research, turbine data sets were recorded and digitized from LM2500 and LM6000 gas turbine engines. Descriptions of the multi-channel data acquisition and hybrid analog/digital anti-aliasing filter system are provided.

A method to incorporate a priori information into the network training set was developed. With knowledge of the network training operation, different sections of the input space could be separated to control the neural classification of those sections.

CHAPTER 1 INTRODUCTION

This research investigates ways that neural network technology can be applied to increase the capabilities of vibration analysis equipment. Vibration analysis is a very subjective science, requiring extensive experience to learn the mechanical meanings of waveforms that are obtained. Neural networks can learn subtle changes in the waveform and thus provide an appropriate analysis tool. The field of neural network analysis was reviewed in order to find architectures best suited for vibration analysis. Of the many possible architectures, the Fuzzy Adaptive Resonance Theory (Fuzzy ART) neural network was chosen for theoretical enhancement to the state of the art. The Fuzzy ART architecture offered advancement to the state of the art in trending analysis, an important component of a machinery vibration-analysis program.

Various neural network architectures were trained to discern between operating modes of the turbine using supervised learning, where the actual mode of operation was presented to the network along with the training data.

The Fuzzy ART neural network was used to build an internal representation of a high-resolution spectrum of the signal. Narrowband changes in the spectrum were reliably discerned using this network. The trending capability provided by this network is believed to be an advancement over previous techniques, in that a large signal space can be analyzed for nonspecific changes. Unexpected changes can be

detected, thus increasing the body of knowledge of turbine operation, by forming a high-resolution picture of the changes in the signal through time.

The principles of machinery vibration analysis and digital signal processing (DSP) were exercised for correct analysis of the vibration signals. Obtaining vibration waveforms presents a problem because both the amplitudes and the form of the signal are important. There are many ways of representing the amplitude in the signal and care must be taken to understand the different units of measurement. For digital signal processing of the vibration signal, all the subtleties involved on obtaining a good signal, including anti-aliasing, must be addressed to obtain a good signal for neural processing.

The application of neural network technology to solve a problem requires at least two areas of concern, the neural network architecture and the subject matter characteristics. Of prime importance is the link between the subject matter and the neural network, otherwise known as feature detection or preprocessing. The choice of neural network is based largely upon the type of information that is to be detected and the characteristics of each neural network. Knowledge of the subject matter is important to determine what effects should be detected and to take advantage of any a priori knowledge in the domain that would help the detection effort. With an understanding of both the neural network and the subject matter, a more effective preprocessing layer can be set up to provide input to the neural network.

Various neural network architectures were investigated to find the most appropriate to process the vibration signals. The Fuzzy ART network was chosen for deeper investigation based upon its properties and due to possible benefits to the field from further analysis of this architecture. The application of neural networks to turbine vibration

also required an in-depth analysis of the properties of the turbine vibration and the existing state of the art in vibration analysis. This research included studying published information, contacting experts in the field, and recording operating turbine waveforms. With detailed information concerning turbine spectral analysis and industry accepted best practices, it became clear that a large portion of the analysis of turbine vibration revolved around trending analysis. Vibration trending involves monitoring for continuing changes in spectrum components with respect to an evolving baseline spectrum. A neural system was developed to perform automatic trending of the detailed wideband spectrum of the turbine vibration. This spectrum was generated to within a resolution of 1 Hz. The spectrum information was wideband in that it encompassed 20 kHz of information. The use of the Fuzzy ART neural net, with appropriate preconditioning and articulation, allowed the entire input spectrum to be monitored to an arbitrary degree of precision. A goal of detecting a 5% change, with respect to full scale, in any spectral component was set and a system developed to accomplish this.

Machinery vibration can indicate the relative health of the machine being analyzed [1], [2], [3], and [4]. The changes in spectral components during extended operation can indicate the wear and possible damage of individual components within the machinery. The current, automated methods used to monitor the vibration of turbomachinery are generally limited to monitoring the amplitude of vibration at the frequency of the rotating element. Other, more elaborate systems can monitor fixed bands of frequencies for deviation from a statistical baseline spectrum. These more advanced systems are generally used only when problems arise. They are also relatively expensive. To perform detailed analysis of the entire spectrum of interest and to recognize

important changes, a human expert was required to visit the installation to view detailed spectrum information and detect problems. Expert consultation is expensive, time-consuming, and not possible for continuous monitoring.

The installation of an inexpensive electronic system to perform extended and continuous monitoring of the machinery should yield longer operating life for the equipment through early detection of machinery degradation. A program of condition-based maintenance (CBM) requires long-term monitoring of the equipment condition and allows the scheduling of maintenance to occur when the system needs it, instead of performing regularly scheduled maintenance. Condition-based maintenance reduces the cost of equipment ownership by performing maintenance on an as-needed basis, generally at longer intervals than specified for scheduled maintenance. CBM is different from run-to-failure maintenance because CBM should be able to detect machinery degradation well before catastrophic damage occurs. Ideally, it should provide a measure of equipment health and remaining equipment length of usability before service. The service operations can also be better targeted to fix detected problems, thus reducing costs and making the maintenance effort more successful.

Trending, where operating characteristics are monitored for continuing changes over long periods, is regarded as one of the best techniques for detecting problems in machinery. A CBM program can use trending to monitor machinery characteristics such as bearing temperatures, filter differential pressures, and vibration spectral feature amplitudes for steadily increasing trends. Increases in the levels of these types of characteristics generally indicate a need for service.

The system developed in this research effort provides an automated method to perform trending of frequency components in vibration signals. The system can monitor the trends of well-known spectral features and can detect and follow new features as they arise in the vibration spectrum. The trending changes were detected using a self-organizing neural network architecture called Fuzzy ART. The operating characteristics of the machinery vibration were learned by the network and thereafter could be continually analyzed to determine if the machinery operating condition matches the conditions at initialization. Changes in the machinery health that were apparent in the vibration would be noted and learned by the neural network for further comparisons. The Fuzzy ART neural network forms an internal compressed representation of the input information to an arbitrary degree of precision. For example, to learn a 32768-point digital spectrum in a method capable of detecting 5% full-scale changes required only 331 neurons. The system does inspect each point in the spectrum for changes, thus providing a very detailed look at the operating conditions.

The neural trending system takes advantage of the Fuzzy ART capability to learn new information quickly and precisely without destroying all the previously learned information or requiring retraining of the entire network to encompass the change.

The new system can be implemented using a single board computer having the appropriate analog input section, thus yielding a somewhat low cost solution that is applicable to widespread installation. The use of this type of system permits a greater confidence level in the operating condition of the machinery, along with providing close monitoring for rapidly developing changes.

CHAPTER 2

GAS TURBINE VIBRATION SIGNAL FEATURES

In the operation of a gas turbine, many potential sources of vibration signals exist due to the complexity of the machinery. There are multiple blade sets, bearings, gearing from attached devices, combustion noise, exhaust noise, and vibrations structurally coupled into the turbine from outside sources. The vibration components from each of these components add into the composite signal obtained from the vibration sensor, or transducer.

Gas Turbine Application

Gas turbines are very complicated machines that operate at extreme conditions of temperature, pressure, and power levels. Tight tolerances are used in the construction of the turbine to ensure correct control of the airflow through the engine and to effectively couple the power of the turbine to external power-transmission devices. The turbine has many stages of rotating blades and stationary stator vanes that serve to compress the air through the turbine. These blade sets generate features in the spectrum that can be individually monitored for changes in amplitude, signifying modification in machinery operation. The gas turbine is also a very expensive device that requires expensive periodic maintenance. Failures in engine components can cause severe and increasing damage if not detected early. Turbine installations generally include vibration sensors and vibration monitoring equipment. The inclusion of the neural trending

system to an existing turbine installation should be straightforward. The only concern would be the frequency spectrum capability of the sensors. The amplitude calibration of the sensors is not as important in the neural trending system but is a concern with other turbine monitoring techniques.

There is at least one rotating shaft within the turbine. Many turbines have multiple aerodynamically coupled shafts, helping the mechanical characteristics, but complicating the vibration analysis. Rows of turbine blades are mounted radially around the shaft. The rows of blades are separated from each other by rows of stationary stator vanes that serve to direct the airflow onto the next set of blades. In some turbines, the stator vane pitch is variable, based upon operating conditions. Prominent features occur in the turbine vibration spectrum at the blade-pass frequencies. For a given stage of blading, the blade-pass frequency occurs at the frequency that is the multiple of the turbine shaft's rotational velocity and the number of blades mounted around the shaft in that stage.

LM2500 Gas Turbine

The General Electric LM2500 gas turbine generates 31,200 shaft horsepower, with output speeds up to 3,800 rpm. It weighs approximately 10,300 pounds and has a power turbine exhaust diameter of 6.7 feet [5]. The LM2500 has many applications, including marine propulsion, offshore platform power generation, gas compression and cogeneration systems.

The LM2500 is a two-shaft engine with variable stator vanes. One shaft is contained within the gas generator (GG) section. A sixteen stage axial compressor and a two-stage, high-pressure turbine are

mounted on the gas generator shaft. The gas generator section also contains the annular combustor for the high-pressure turbine. The other shaft is contained within the six-stage, low-pressure power turbine (PT) section. The power turbine is aerodynamically coupled to and driven by the exhaust from the gas generator.

The manufacturer's specified vibration monitoring system only analyzes the once-per-rev vibration, which is the vibration occurring at the frequency of rotation of each of the shafts. There are many more components in the spectrum that can be detected and used for condition-based monitoring. Most of these components are related to blade-pass frequencies. In cases of damage or wear, single turbine stages may show increased vibration. While this may affect the once-per-rev vibration level, this effect is indirect. A more direct way to sense damage in the individual blade stage would be to monitor the vibration spectrum feature related to this blade stage.

Data Collection

Turbine data sets were recorded on a Kyowa Dengyo RTP-650A multichannel data recorder. This data recorder uses frequency modulation to record the voltages present on the input channels onto Beta-type videocassette recorder tapes. The data recorder was operated at a tape speed of 38.1 cm/sec to obtain a wide bandwidth (DC-20kHz) and large signal-to-noise ratio (48 dBrms).

LM2500 data. LM2500 turbine engine data sets were recorded at the General Electric Aerospace Engine Group facility in Cincinnati, Ohio. These data sets were recorded in the turbine test facilities from tapes that contained different operating modes of the LM2500 turbine. In marine applications, the LM2500 turbine is operated over a

wide range of speeds, with numerous start-ups and shutdowns. To obtain a representative set of turbine vibration information required the recording of accelerations, decelerations, start-ups, and steady state operation.

LM6000 data. LM6000 turbine engine data sets were recorded at the Florida Power Cogeneration Plant located on the University of Florida campus in Gainesville, Florida. These data sets were recorded from an operating turbine. The signals were obtained from the raw data outputs of a Bentley-Nevada vibration monitor. In power-generation applications, the LM6000 is generally operated over a narrow range of operating conditions to provide a steady frequency and voltage output from the generator. Different power demands cause changes in the turbine operating point. The data sets were recorded during late afternoon and early morning to obtain multiple operating points.

Digitizing equipment. A digital signal processor (DSP) based system was assembled to digitize the multichannel vibration data. This system consisted of

1. Spectrum Signal Processing PC/C31 DSP board with AM/D16SA ADC/DAC Daughter Module. (This board contained a 40 MHz Texas Instruments TMS320C31 DSP chip and 128K of memory. The daughter module contained two Burr-Brown 16-bit analog-to-digital converters.)
2. Spectrum Signal Processing PC/16IO8 Multichannel I/O board. (This board contained sixteen 12-bit analog-to-digital converter channels.)
3. Dell dual-Pentium Pro computer with 128 MB of RAM running Windows NT.
4. Lockheed Martin Corp. Dual-channel anti-aliasing filter.

The Texas Instruments C-compiler was used to create code to operate the digitizing equipment. The digitizing equipment and the programs that were written for the DSP card and PC to perform the digitization are discussed further in Appendix A.

Tape survey. The data sets that were obtained from the turbines were reviewed to determine where the important characteristics were located on the tape. To perform the survey, the entire contents of each tape were digitized into the computer and a spectrogram was taken of each set of data. The tapes were recorded in both the forward and the reverse direction, on different channels. The terminology used to describe the spectrogram images indicates the signal name, the tape number, and tape direction.

Each tape contains a large amount of data. To obtain a file with a recording of a complete tape required that the input signal be filtered to reduce the number of samples that must be recorded. The turbine once-per-rev rotational vibration is all contained in the frequency range from near-DC to 200 Hz. To obtain this range, plus the first two harmonics of the once-per-rev vibration, a bandwidth of DC to 600 Hz was recorded. A pair of fourth-order Butterworth analog anti-aliasing filters was constructed. The input vibration signals were filtered by the anti-aliasing filter and then four-times oversampled in the digitizer at $f_s = 5003.256$ [samples/sec]. The DSP card was programmed to execute a 35th order IIR digital filter and the final signal downsampled, or decimated, to a effective sampling rate of $f_s = 1250.814$ [samples/sec], having a Nyquist rate of $f_{NY} = 625.407$ Hz. These rates were selected because they could be obtained to a high degree of precision using an integer division of the sample-clock oscillator of the DSP card. The high-order digital filter was required to remove a

large energy structure at approximately 800 Hz that was due to turbine combustion noise. The anti-aliasing filter system is discussed in Appendix B.

Spectrograms from some of the LM2500 and LM6000 turbine engine tapes are presented below. The figures show the gas generator (HP turbine for LM6000) and power turbine (LP turbine for LM6000) vibration spectrograms, along with the frequencies derived from the tachometers showing the speed of the gas generator and the power turbine. The tachometer signals can be compared with the energy traces in the spectrograms to see what generates the different energy traces. The highest power traces tend to come from gas generator once-per-rev vibration and this gas generator vibration yields the most harmonics. In the figures, there are periods where there appear to be missing or garbled data sections. These periods of data loss can occur when tape calibration information is recorded, when the source tape had stopped, or during possible tape damage. The tachometer waveforms also display a hunting action during these periods of tape loss. The LM2500 tachometer signal is derived from a sine wave that is a multiple of the rotational velocities of the turbines, 47 times the gas generator speed and 83 times the power turbine speed. This required a higher sampling rate than the velocity data in order to capture the tachometer data, multisampling each tachometer waveform during each sample of the vibration data. The tachometer data sets were converted to frequency using a zero-crossing counting mechanism. The implementation of the tachometer is described in Appendix A.

LM2500 data. The following groups of spectrograms and tachometer plots show the evolution of some of the features in the LM2500 low-frequency spectrum. Multiple harmonics and features related to both the gas generator and power turbine shafts can be seen.

The Tape 1, forward, as shown in Figure 2-1, Figure 2-2, Figure 2-3, and Figure 2-4 shows strong features. There is a startup event occurring between approximately 75 seconds and 120 seconds. At least three harmonics of the gas generator are visible in this bandwidth. The power turbine vibration is faintly visible. Following this initial startup transient, the turbine settles down at idle for the remainder of the tape. The characteristics of the power turbine and gas generator vibration are nearly identical. Some minor differences are visible. The gas generator accelerometer is closer to the high-pressure turbine of the LM2500 than is the power turbine accelerometer. The gas generator turbine blades under more stress as they are involved in the combustion, compression, and acceleration of the air mass through the turbine. For this reason, the gas generator waveforms will be used in the analysis.

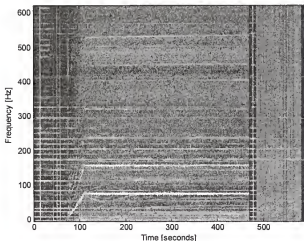


Figure 2-1. LM2500 tape one, forward, gas generator vibration.

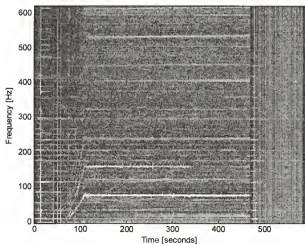


Figure 2-2. LM2500 tape one, forward, power turbine vibration.

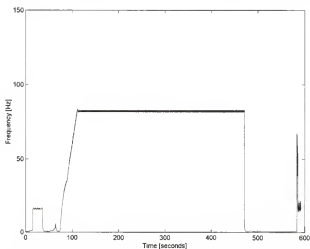


Figure 2-3. LM2500 tape one, forward, gas generator speed.

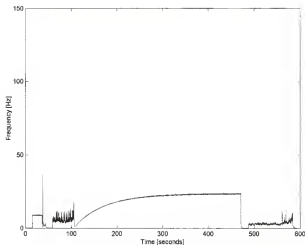


Figure 2-4. LM2500 tape one, forward, power turbine speed.

Tape two, forward, as shown in Figure 2-5, Figure 2-6, Figure 2-7, and Figure 2-8 shows steady operation at high power. Many strong harmonics are visible. This tape had recorded the turbine operating in a high-power mode. The turbine components were eliciting more vibrations in the spectrum due to the high-power operation. Therefore, signals from this tape were used in the development of the Fuzzy ART neural network application to be described in Chapter 4. The section of tape centered around 300 seconds was used for initial analysis because there appeared to be no transients in the spectrogram at this time. Variance in the spectrum due to turbine operation was thus minimized. When multiple spectrums were used in the analysis, they also came from this tape, but more spread out in time.

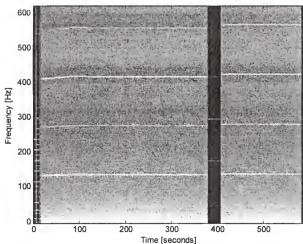


Figure 2-5. LM2500 tape two, forward, gas generator.

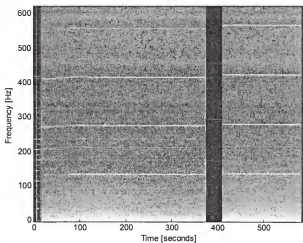


Figure 2-6. LM2500 tape two, forward, power turbine.

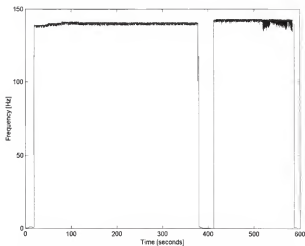


Figure 2-7. LM2500 tape two, forward, gas generator speed.

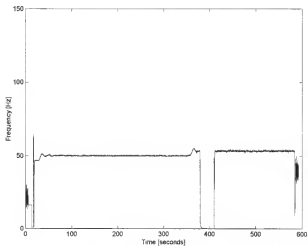


Figure 2-8. LM2500 tape two, forward, power turbine speed.

Tape three, forward, as shown in Figure 2-9, Figure 2-10, Figure 2-11, and Figure 2-12 is the most dynamic of all the recorded tapes. It shows relatively high-power operation (15-100 [sec]), with a spin-down. During the spin-down, the first and second harmonics of the power turbine speed is apparent in the spectrograms. This is a very crowded spectrum, making it difficult to separate components from the two turbine sections. The turbine sections are not mechanically coupled and are thus free to spin at different rates. While it may be possible to roughly predict the actual turbine speeds, based on load, fuel type, air pressure and temperature, and other factors, the actual speeds are not linked with a steady mathematical formula. This requires neural net training to be performed on the actual installation, with factors governing the turbine performance used as input to the network. Many modes of operation must be learned to allow dynamic tracking during acceleration and deceleration of the turbine. When the operating mode of the turbine matches one of the modes that have been learned, then a time sample of the turbine operation will be obtained and analyzed. The power turbine tends to change speeds more slowly than the gas generator because the power turbine is generally driving large masses through direct attachment. The gas generator is driven primarily by thermodynamic forces obtained through the burning of fuel and is not coupled to large masses. Ideally, the condition monitoring will take place at steady speeds, but dynamic operation can be monitored through controlled slow ramps in speed. In application, periodic checks can be made by having the vibration analyzer provide control signals to the turbine controller to take the turbine through a range of operating points and dynamic operation. This would help ensure that the turbine activates the various learned operating modes. While this controlled check may be suitable for maintenance situations,

online dynamic condition monitoring must be provided for general usefulness. The online monitoring will depend on learning the various operating modes that are seen in the application. This would require initial installation efforts to align the monitoring equipment. Any automation capabilities that can be introduced to provide this initial alignment will greatly enhance the applicability of the device. The alignment will be required both during initial installation and after equipment overhauls or maintenance. If the system could run in a learning mode and determine novel modes of operation, then suitable long-term memory traces could be generated to cover the expected modes of operation.

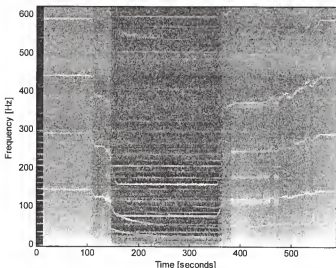


Figure 2-9. LM2500 tape three, forward, gas generator.

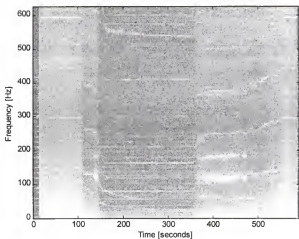


Figure 2-10. LM2500 tape three, forward, power turbine.

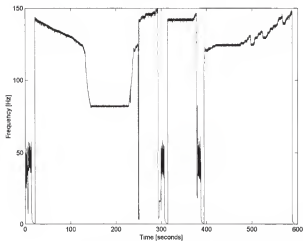


Figure 2-11. LM2500 tape three, forward, gas generator speed.

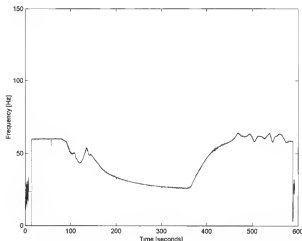


Figure 2-12. LM2500 tape three, forward, power turbine speed.

LM6000 data. The following diagrams, Figure 2-13, Figure 2-14, Figure 2-15, and Figure 2-16 show spectrograms related to two operating modes of an LM6000 turbine. The frequency and time axis are different for these figures. The figures were provided to illustrate some of the characteristic of the turbine. These spectrograms were based on the high-speed sampler and showed the presence of features at higher frequencies. These features are likely related to blade-pass energy, as will be described in the turbine features sections. The LM6000, when used in a power generation application, generally provides a very stable operating condition. This is very inviting for a first application of this new technology because the broad dynamics of the LM2500 in a propulsion application are not evidenced. The LM6000, in a power generation application, should thus require less memory to represent the operating spectrum. Changes in the spectrum should also

be more readily interpreted as machinery wear, as opposed to artifacts of propulsion system interconnections and effects induced by relatively rapidly-varying operating conditions.

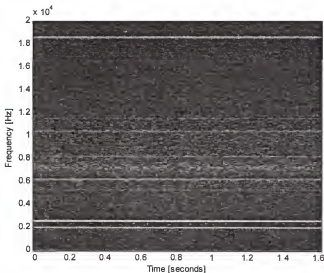


Figure 2-13. LM6000 high-pressure turbine, lower power.

Vibration Transducer

Different types of transducers are used for vibration detection, including noncontact displacement transducers, velocity pickups, and accelerometers [6] [5]. The transducer used for the analysis must have sufficient bandwidth to encompass the primary features of analysis, in this case including turbomachinery blade-pass frequencies in the kilohertz range. The transducers must also be able to tolerate high temperatures and a relatively harsh environment.

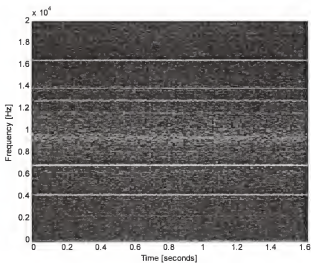


Figure 2-14. LM6000 low-pressure turbine, lower power.

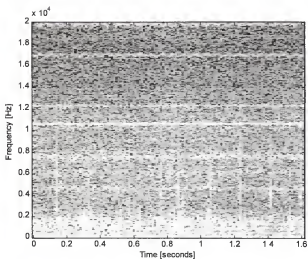


Figure 2-15. LM6000 high-pressure turbine, higher power.

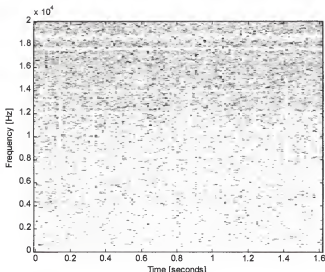


Figure 2-16. LM6000 low-pressure turbine, higher power.

Noncontact displacement transducers operate with eddy current sensing of the amplitude of imposed electromagnetic waveforms in the metal of the rotating shaft. These types of transducers provide good low frequency performance, but are generally limited to a maximum frequency of 1 kHz to 1.5 kHz.

Velocity pickups operate by suspending a ferrous material inside a tube wrapped in magnet wire and measuring the vibration as changes in the magnetically induced voltage that appear across the coil. These types of transducers can reproduce a maximum frequency of approximately 4 kHz, but have poor low frequency characteristics. Velocity pickups also have a drawback in that they are mechanical devices with moving parts, thus being unsuitable for installation in a demanding environment.

The spectral components associated with turbomachinery blade-passing frequencies can extend into the tens of kilohertz. The highest frequency component that is expected in the LM2500 turbine spectrum is 23200 Hz, based upon the number of blades and the maximum turbine rotational velocity.

Accelerometers offer the best characteristics for high-speed vibration sensing [8]. They can be constructed using a small mass connected to piezoelectric material that generates a varying charge as the mass reacts to the vibration of the turbine. These sensors have no moving parts, can withstand high temperatures, and can be used up to approximately 50 kHz. These transducers are the best choice for this turbine analysis application and are generally installed on the LM2500 and LM6000 gas turbines. Accelerometers produce a small output voltage proportional to the stress caused by the mass on the crystal when subjected to vibration. The accelerometers used in this research are electronically integrated in a charge amplifier to produce outputs proportional to velocity. The frequency spectrum of these accelerometers begins to taper off in amplitude as the frequencies extend above 10 kHz, capturing the blade-passing frequencies.

Tachometers are also important in vibration analysis because the rotational velocity of the turbine is needed to determine the frequencies of many other synchronous features such as the blade-passing vibrations. The LM2500 tachometers generate a signal by magnetically sensing the passing of gear teeth on a gear driven by the turbine shafts. The resulting signal represents the turbine speed multiplied by the gear ratio. Highly accurate detection of the speed is possible through digital processing of the gear tooth waveforms.

Neural Network Features

To apply neural networks to a problem, inputs must be derived that characterize the problem to the network. These inputs are called features. The set of features must be expressive enough to allow the neural network to distinguish between the operating modes to be recognized. To help keep the neural net simple and effective, the features should be orthogonal or at least non-redundant [9]. Different features should provide different types of information concerning the problem. The features that were obtained from the turbine vibration signal are presented below.

The raw signals from the accelerometers must be conditioned before use in the neural network for dimensionality reduction and tailoring of the decision regions of the neural network. For a digital neural network, the signals must be converted into digital signals. An analog anti-aliasing filter is needed to ensure that the analysis bandwidth of the digital signal contains no artifacts of digitization caused by frequencies higher than the Nyquist rate (sampling frequency / 2) reflecting as lower frequencies. The anti-aliasing filter built for this research is described in Appendix B. The filtered signal must be presented to an analog-to-digital converter system. The digitizer system created for this analysis provided two channels of 16-bit resolution, for obtaining signals from both the power turbine and the generator sensors. The digitizer system simultaneously captured two tachometer inputs using 12-bit resolution. The sampling rate for the vibration signals was 40026.06 [samples/second] and the tachometers were sampled at 20013.03 [samples/second]. The digitizer was constructed using commercial digital signal processing equipment, programmed in C and assembly and was used to capture signals ranging

from 1.5 megabytes (MB) to greater than 20 MB in size. The digitizer system is described in Appendix A. The each vibration data point was processed and stored as a 32-bit binary floating-point to avoid integer word-length problems associated with the dynamic range. Once the signals were digitized, they were processed in Matlab to extract features for the neural network processing. The neural nets were constructed both in C-language code and in NeuralWorks Professional II/Plus from NeuralWare, with the network receiving primary analysis constructed in C.

Time Domain Features

Vibration analysis may be considered to consist of the study of the spectrum, but the time domain signal can provide some useful information. The time domain signal can show if transient impacts are occurring, or if there is a problem with the sensor system itself.

The most basic time-domain features are the levels of the signals in terms of the amplitude descriptors: average, peak-to-peak, and root-mean-square (rms) values. In some vibration analysis publications, these values are shown as being related to one another by a simple multiplication. A simple multiplication applies only when the signal is a simple sine wave. In signals that are more complex, the time-domain features must be computed separately as shown below. For instance a narrow peak, resulting from a missing gear tooth, in a signal will affect the peak-to-peak value, but will not affect the rms value to a great degree.

Average. In a vibration monitoring sense, the average value of the signal is obtained by summing the absolute values of each sample in

the signal and dividing by the number of points in the signal, as follows:

$$\text{average}(x(n)) = \frac{\sum_{n=1}^N |x(n)|}{N} \quad (1)$$

Peak-to-peak. This is the maximum value found in the signal minus the minimum value in the signal, as follows:

$$\text{peak-to-peak}(x(n)) = \max(x(n)) - \min(x(n)) \quad (2)$$

The peak value is $\frac{1}{2}$ the peak-to-peak value. Peak or peak-to-peak levels of the signal indicate the maximum excursions in amplitude of the signal. It is useful in ratios with other signal level measures, such as in the crest factor, to detect waveform changes indicating higher impact noise.

Root-mean-squared (rms). Root-mean-squared calculations provide a measure of the energy level in the signal. The rms value is the square root of the mean of the sum of the squares of each point in the signal, as follows:

$$\text{rms}(x(n)) = \sqrt{\frac{\sum_{n=1}^N x(n)^2}{N}} \quad (3)$$

Crest factor. The crest factor is derived from the basic amplitude levels of the signal. The crest factor is defined as the peak value divided by the rms value. The crest factor increases as machinery wears, shown by larger transients with respect to the rms value. A problem with crest factor occurs as a defect begins to spread in the spectrum. As the defect spectral feature spreads, the rms value increases more quickly than the peak value and the crest factor will decrease.

Form factor. The form factor is defined as the rms value divided by the average value.

K-factor. The k-factor is defined as the peak value multiplied by the rms value [5]. This allows defects to cause an increase in the K-factor even if the defect spectral feature widens. The problem described in the crest factor discussion is thus alleviated with the use of the K-factor amplitude descriptor.

Kurtosis. Kurtosis is a measure of how outlier-prone the distribution of amplitudes in the signal is [10]. Kurtosis [11] is the normalized, fourth central moment of the input time-domain signal, having the following form:

$$\text{kurtosis}(X) = \frac{E(X - \bar{X})^4}{\sigma^4} \quad (4)$$

where

$\bar{X}$ = mean value of signal

σ = standard deviation of signal.

Computing the kurtosis of a digital signal requires the following manipulations:

$$\text{kurtosis}(x(n)) = \frac{\left(\frac{1}{N} \sum_{k=1}^N \left(x(k) - \frac{1}{N} \sum_{i=1}^N x(i) \right) \right)^4}{\left[\frac{1}{N} \sum_{j=1}^N \left(x(j) - \frac{1}{N} \sum_{i=1}^N x(i) \right) \right]^4}$$

The normal distribution has a kurtosis of three. A kurtosis level greater than three can indicate machinery wear because larger transients are occurring. These transients may be caused by bearing wear, as impacts begin to occur in the bearing raceway due to non-circular bearings. Kurtosis is related to the crest factor and provides a good measure of the wear of the machinery when used in a trending system [12].

Frequency Domain Features

A large number of new features are obtained when the time-domain signal is transformed into the frequency domain. In DSP, this transformation is usually performed with the Fast Fourier Transform (FFT). The Wigner-Ville Distribution was successfully used for time-frequency distributions in vibration analysis [13] in an earlier effort. The Wigner-Ville Distribution is well suited for analysis of nonstationary signals as may be found in gas turbines used in propulsion applications. Among its capabilities include the ability determine the energy in certain bands of the spectrum through integration and accurately tracking frequencies that are changing in time. A drawback that limited its applicability in some of the neural applications is the generation of cross-term energy that shows up in the spectrum as artifacts, without physical cause. Certain techniques were investigated to allow its applicability. In this analysis, the FFT was used to lower the complexity of the resulting system, concentrating on the neural aspects of the problem. The Wigner-Ville Distribution can likely provide many benefits to monitoring the signal changes during acceleration and deceleration.

To view all the features in the high-resolution spectrum, a full-length printout of a sample spectrum was created, requiring 17 sheets of 8½" x 11" paper, printed in landscape, to print the entire spectrum. This expanded printout permitted a detailed analysis of the vibration components with respect to the expected features of the turbine. Blade-pass frequencies and turbine harmonics were identified by this method.

Turbine tachometer signals. The turbine shaft speed is captured with a tachometer and used for control and vibration analysis of the

turbine. The turbine speed is used to determine the power range that the turbines are operating in and to help determine the frequencies of other features. Many spectral features are set at fixed multiples of the turbine speed, making it crucial that the tachometer be accurate. In the commercial-of-the-shelf (COTS) input digitizer hardware, dedicated tachometer hardware was not available to directly obtain precision tachometer values. Careful analysis of real-time rates, system limitations, and DSP algorithm design was needed to obtain accurate tachometer readings for this analysis. The resulting tachometer system provided the needed accuracy. The tachometer system is described in Appendix A.

High frequency spectrum. Two bandwidths of signals were recorded for this analysis. A high-frequency bandwidth, extending from DC (Direct Current, 0 Hz) to 20013.03 Hz and a low-frequency bandwidth extending from DC to 625.71 Hz. The precision in the frequency representation is needed for correct interpretation of the high-resolution bandwidth and is easily provided by the crystal clock of the digitizing equipment. The high-frequency spectrum proved to be the most useful in the neural trending system because it contained the turbine blade-pass frequencies and their harmonics.

Low frequency spectrum. The low frequency spectrum had a range of DC to 625.71 Hz, encompassing the first two harmonics of the turbine rotational velocity. The turbine rotational velocity is also known as the 1X speed, signifying one times the speed of the turbine. Much analysis is possible in this range, including 1X, 2X, and 3X level sensing, and subsynchronous (less than 1X but proportional to the 1X speed) features such as oil slosh and rotating stall.

One-per-rev vibration. The once-per-rev speed is the speed that the turbine is rotating, also known as the 1X speed. High levels of 1X

vibration are indicative of many faults, especially turbine imbalance. In the LM2500 dual-shaft engine, there are two 1X speeds, that of the power turbine and that of the gas generator. These spectral features are shown in Figure 2-17, along with their corresponding harmonics and exhaust rumble.

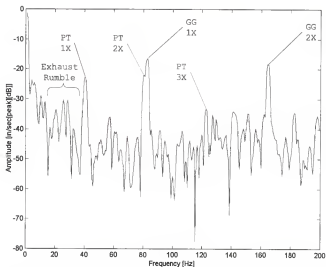


Figure 2-17. Turbine once-per-rev frequencies (1X) for power turbine (PT) and gas generator (GG), with harmonics (2X and 3X) and exhaust rumble.

Blade-pass frequencies. Each blade set on a turbine shaft contains a number of blades, evenly spaced around the shaft. The spectrum shows an energy feature related to each blade set, at the frequency: $f = B \cdot f_n$, where B is the number of blades in the blade set and f_n is the 1X rotational speed of the turbine. An increase in amplitude at a blade-passing frequency can indicate blade cracking, warping, pitting, or a change in clearance of the blade edge with the

turbine enclosure. An important blade-passing feature is shown in Figure 2-18. The 11th to 16th blade sets all have the same number of blades and thus combine to form a large feature at 76 times the gas generator 1X frequency.

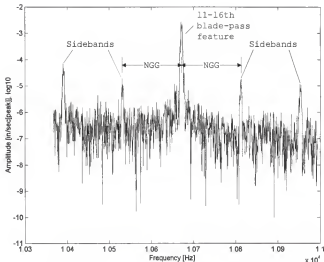


Figure 2-18. A blade-pass feature and sidebands.

The analysis of the turbine blade-passing features required cataloging of all blade-stages and the number of blades in each stage. The turbine plan view, presented in Figure 2-19, shows the layout of the LM2500 turbine. A Vibration Test and Analysis Guide, VTAG, was developed for the LM2500 engine, based upon documentation from the manufacturer, of which some illustrative data points and values are reproduced in Table 2-1. Data sets were obtained for all the turbine blade sets and used for analysis in the neural networks.

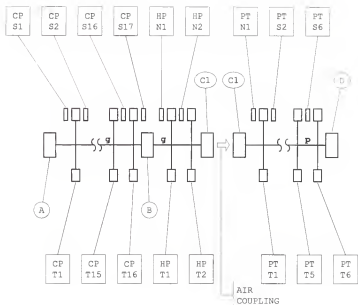


Figure 2-19. LM2500 gas turbine plan view.

Sidebands. Sidebands are spectral features arising to both sides of another feature in the spectrum. Sidebands arise around the blade-pass features due to modulation with the engine speed of rotation.

Increases in sideband levels around the blade-passing frequencies of a gas turbine can be attributed to higher power operation, rotor damage, or a broken blade. These sidebands are separated from the main feature by the 1X frequency of the turbine. In the LM2500 turbine with two shafts and two 1X frequencies, the sidebands always appeared with respect to the gas generator 1X speed (also called NGG).

An excerpt of the high-resolution data is shown in Figure 2-18, centered around the gas generator 11th through 16th blade-stage spectral feature. Note the two sidebands to each side of the main feature.

These are separated from the main feature by multiples of the gas generator speed.

Table 2-1. Vibration test and analysis guide (VTAG) data.

ITEM	DESCRIPTION	# ELEM- ENTS	FREQ.
g	Gas generator shaft (reference)		1.00g
p	Power turbine shaft (reference)		1.00p
CP S1	1 st stage compressor stator blading	35	35g
CP T1	1 st stage compressor turbine blading	36	36g
CP S2	2 nd stage compressor stator blading		
CP T15	15 th stage compressor turbine blading	76	76g
CP S16	16 th stage compressor stator blading		
CP T16	16 th stage compressor turbine blading	76	76g
CP S17	17 th compressor stator blading set		
HP N1	1 st stage high pressure turbine nozzle		
HP T1	1 st stage high pressure turbine blading		
HP N2	2 nd stage high pressure turbine nozzle		
HP T2	2 nd stage high pressure turbine blading		
PT N1	1 st stage power turbine nozzle		
PT T1	1 st stage power turbine blading	166	166p
PT S2	2 nd stage power turbine stator blading		
PT T5	5 th stage power turbine blading		
PT S6	6 th stage power turbine stator blading		
PT T6	6th stage power turbine blading		
A	Bearing		
B	Bearing		
C1	Bearing		
C2	Bearing		
D	Bearing		

Noise elements. There are some wideband noise elements in the turbine spectrum, including exhaust rumble and combustion noise. The exhaust rumble occupies the spectrum between approximately 22 Hz to 31 Hz, as shown in Figure 2-17. The combustion noise occupies an area in

the spectrum between approximately 800 to 900 Hz. The combustion noise has a magnitude nearly 40 dB higher than the noise floor. Due to the proximity of the high-energy combustion noise to the low-frequency spectrum bandwidth of 0 Hz to 600 Hz, care had to be taken with the design of the anti-aliasing filter. The anti-aliasing filter that was constructed was a hybrid analog/digital filter with 4X oversampling, as described in Appendix B. The magnitude of combustion noise was useful in the mode-detection neural network to indicate accelerations.

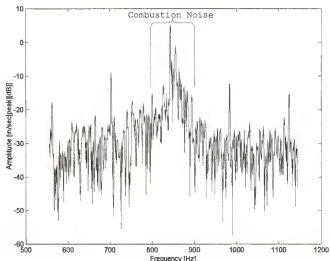


Figure 2-20. LM2500 gas turbine combustion noise.

Entrapped oil. When oil builds up inside a turbine element, it causes vibration when the turbine is rotating. The entrapped oil rotates with the turbine element, but at a lower, or subsynchronous, speed. Oil in the bearings causes oil whirl, where a wedge of oil whirls around the inside of the bearing. Oil whirl shows up in the

spectrum between 40 to 48% of the 1X speed. Oil whirl can be caused by excessive clearance, or insufficient radial loading in the bearing. When the entrapped bearing oil is excited at a critical shaft speed, the oil whirl can turn into oil whip. Oil whip is very destructive, causing violent vibration. Oil whip can be detected as subsynchronous vibration during operation at a shaft critical speed and verified by increased amplitude at this frequency during a turbine coast down. Oil whirl is proportional to the speed of the turbine, oil whip, when begun, will remain at a constant speed corresponding to a critical shaft speed.

Another oil phenomenon identified in the LM2500 is oil slosh. In this case, oil seeps through the oil seals in the bearings and becomes entrapped in the interior of the turbine rotor assembly. A large amount of oil can be entrapped within the rotor, causing a very large vibration [14]. Oil slosh is seen in a range of frequencies depending on the amount of oil, but generally noticed in a distribution around 91% of the 1X speed of the power turbine (the NPT speed).

Rotating stall. A full aerodynamic stall condition in a turbine blade-set can cause severe damage. Precursors to the full stall of the blade set include rotating stall conditions, where a packet of high turbulence air rotates around the blade sets in the turbine [15]. Rotating stalls can be seen during acceleration of the turbine. They appear as a fluctuating vibration from a physically stalling stator vane going in an opposite direction of rotor rotation around the case. The stall cell moves from stator vane to stator vane at a frequency related to the flexure of the stator vane. As the turbine passes through certain frequencies in its spin-up, rotating stall energy may be seen at 47% of the 1X speed of the gas generator.

CHAPTER 3 NEURAL NETWORK SELECTION

The implementation of artificial neural networks began at least as early as 1957, with Frank Rosenblatt's creation of the Mark I Perceptron. Early papers on neurocomputing extend back to the early 1940s with McCulloch and Pitts' research into neural network computing capabilities. With the advent of more capable and widespread computing capability and after a lull in the early 1980s [16], research into neural network (NN) technology has been increasing dramatically. After the development of back-propagation learning, more complicated problems began to be solved with NNs. Many classes of NNs, and modifications to existing types, have been generated both to improve performance and to accommodate different types of problems. In general, neural networks are composed of many identical processing elements, having a large number of interconnections, that self-adjust to learn input information. The model for artificial neural networks is the way in which the brain is believed to perform certain functions or tasks [17]. The brain contains cells called neurons that are interconnected in complex ways. Artificial neural networks attempt to model brain processes by interconnections of artificial neurons. These artificial neurons attempt to imitate some of the processes believed to occur in physical neurons. The individual neurons perform very little processing alone. Their combined effect in a network provides the interesting properties associated with neural networks, such as learning and classification [18] [19]. The physical neurons have a

cell body, represented by the processing element in artificial neurons. Physical neurons also have axons, synapses, and synaptical activations, represented by the interconnections and weights of the artificial neurons. The neurons in the brain are believed to communicate and generate thoughts or responses through electrochemical communications in response to a stimulus. The combined stimulus into each neuron increases the activation level of that neuron, until the activation is high enough to cause the neuron to emit a pulse to the other neurons to which it is connected. Artificial neural networks simulate this action using summing junctions, weights and activation functions to group the inputs and provide a change in output with sufficient input activation.

There are many variations of the activation functions and operation of the individual neuron. Even more so, there are many variations in the architecture governing the interconnections of the neurons, their training, and their use. A popular architecture, the multi-layer perceptron, groups its neurons together in layers and interconnects the output of each neuron in a layer to the inputs of every neuron in the next layer. This can be modified to cause interconnections to skip layers, or to cause the outputs to wrap back around to the inputs. New types of neural networks, and especially systems of multiple neural networks, will certainly be introduced, while the existing ones are put into more applications.

The type of training that is applied to a neural network provides a basic classification between the many types of neural networks. A network can be trained in a supervised method, or in an unsupervised method. In supervised neural network training, the training input to the neural network includes a supervisory section that indicates the output that the neural network is expected to produce. In a neural network trained with an unsupervised learning method, also called a

self-organizing system, the network analyzes the input data using the properties of the neural network and the input data, without external classification provided in the training set. The network trained with an unsupervised learning method attempts to represent the statistical properties of the input data [17]. Of the various types of neural networks are many implementations that serve different purposes, with different strengths and weaknesses. It is important to choose a network that is appropriate to both the problem and the data available to the problem space.

Another neural net classification of the network uses feedback of network output to network inputs. This is called a recurrent network. This distinction is thus between static networks, not using feedback, and temporal networks, that use feedback or another method to incorporate a time-dependency into the network operation. Temporal networks, such as the Gamma Model [20], the Real-Time Recurrent Learning algorithm [21], the Space-Time Neural Network [22], and the Leaky Integrator Neuron [23], provide an ability to learn and monitor a time-domain process such as a signal waveform. An application of temporal neural networks to turbines would be in the monitoring of individual turbine blades using pyrometers embedded in the turbine case near blade stages. As the individual blades pass the pyrometer, the temperature pattern of the blades could be measured and compared to a learned waveform. This could allow the turbine to operate safely at a higher temperature, thus increasing the turbine efficiency. Temporal networks were not investigated in this research because of the highly complicated nature of the turbine spectrum that seemed amenable to bulk statistical techniques including time-domain moments and estimations of the frequency spectrum of the vibration. Temporal networks are valuable, and further research into their applicability would likely

provide many benefits, including transient detection. They could provide a quick monitoring of the instantaneous turbine waveform, but the turbine cannot be halted instantaneously, thus reducing the benefit of the high-speed defect detection. Temporal networks would likely provide benefit in detection of impact noise that can be seen occurring in the time-domain waveform itself. They may also be very important in analysis of the trending data, monitoring the long-term changes over time.

This research investigated the applicability of a variety of neural network architectures to the problem of detecting machinery defects evident in the machinery vibration spectrum. Both supervised and unsupervised network architectures were investigated. For a supervised network to be effectively trained, the network should be presented with examples of data containing the different characteristics that are expected to be distinguished, along with supervisorial data sets that indicate the classification of the input data. Because an expensive gas turbine was used to supply the vibration data for this research, it was difficult to obtain vibration data containing defect responses. Thus, the investigation into the supervised neural networks was limited to a set of tests to determine the operating mode of the turbine, as groundwork for further research if data containing defect responses becomes available. The unsupervised network yielded results that are more promising. The unsupervised network was investigated in depth because it did not require data from defective turbines in order to operate successfully. The unsupervised network forms internal representation of the turbine spectrum, from which it can detect changes that possibly signify turbine damage. The Fuzzy Adaptive Resonance Theory (Fuzzy ART) unsupervised neural network was applied to create a network capable of

detecting small changes within the turbine vibration spectrum. Thus, the major concentration of this research involved the unsupervised network architecture to create a new vibration analysis tool and provide a contribution to the field.

The investigation into the supervised network consisted of a survey of the performance of the various networks with the available data concerning turbine operating modes. This was performed using the NeuralWorks network development and prototyping tool, which provided a large assortment of modern and historical networks for application to the problem. The unsupervised network, Fuzzy ART was hand-coded in C, to provide a robust means of experimentation and detailed research into the network parameters and operation, providing the main thrust of the research.

Supervised Neural Networks

To use a supervised neural network, it is necessary to be able to provide the supervisory data. To train a supervised network to distinguish between damaged turbines and good turbines would require data from both good and damaged turbines. In the case of an \$8M gas turbine, it is difficult to obtain failure data. Even to repair the compressor from damage caused by ingested metal or stall conditions costs \$1.2M, so inducing damage for data collection is very expensive. There is limited availability of recorded vibration data and pre-recorded data of a failing turbine is even scarcer. Turbines in need of repair are also generally not run, thus hindering data collection. At the time of this research, the only operating turbines known by the manufacture (General Electric) to be in need of possible repair were either on a Navy ship or in Cartagena, Columbia. Attempts were made to

get on the Navy ship, but it proved unworkable due to their mission. Bruce Thompson, the Navy vibration analyst who did go to the ship, provided information on the types of problems he has encountered on different LM2500 gas turbines, thus influencing this research effort.

Supervised neural networks were analyzed using the available data to determine the operating mode of the turbine. Different preprocessed input features were used to increase the accuracy of the neural outputs. These networks analyzed the vibration to determine the operating mode of the turbine, whether operating at high power, low power, accelerating or decelerating. Although these operating modes could be easily determined directly from the tachometers, the ability to discern operating modes was used to test the viability of different neural networks, and their input feature sets. The tachometer information was used to create the supervisory data sets for training. It is expected that the network architecture best suited to distinguishing operating modes would also provide similar capabilities detecting defect responses in turbine vibration, if these data sets become available. The features supplied to these supervised neural nets were described in Chapter 2 and are measures of the vibration strength, from both time-domain and frequency domain analysis, and a measures of time-domain amplitude distribution. These features are indicative of the stress put on the turbine in the different modes. While useful in discerning operating mode, they should also be valuable in detecting defect responses in the vibration signal.

Because of the unavailability of defect response data, the investigation into supervised neural nets was limited to comparisons of the ability of different network structures to detect turbine operating modes. This investigation still required data acquisition, preprocessing, and neural network system construction.

Unsupervised Neural Networks.

According to turbine vibration experts from the Navy and industry, monitoring the trends of changes in the spectrum is one of the most important vibration analysis techniques. A vibration component that has steadily increased in amplitude over time can indicate turbine deterioration related to an associated mechanical component. An unsupervised neural network architecture can be used to detect changes for use in a trending system.

Neural networks form an internal representation of the input data. This representation provides an automatic classification of the data. This internal representation can then be used to perform novelty detection, where cases that differ from the previously learned information can be detected.

The ability to detect changes helps create a valid system when the characteristics of defective turbines are not known. Since the defect response data is unavailable, the network is trained to recognize a good turbine and to use the novelty detection feature to indicate when the input spectrum no longer looks like that of a good turbine. This technique allows a neural trending system to be created that will automatically detect changes anywhere in the spectrum. The neural network compares new information to its internal representation of the turbine vibration characteristics to detect changes. Through theoretical analysis of the Fuzzy ART unsupervised network, the parameters, architectural configuration, and preprocessing required to achieve desired performance were determined and tested.

Neural networks that form crisp boundaries around the classification regions enhance the ability to detect novel inputs. Some neural network architectures generalize to produce an output for

any value from the input space. For instance, multilayer perceptrons with continuous activation functions also produce a continuous output [17]. Networks that produce output values in a continuous manner in the neighborhood of previously learned data are very good for function estimation, but perform poorly at detection of novel data. Other networks have neural regions that do not generalize over the entire input space, but only recognize data that is very close to the data used in its training. Thus, the network may not recognize an arbitrary input. While this may be considered over-training for a function-estimation application, this is a good property for detecting changes in the turbine for use in trending.

The ability to process vibration data using the Fuzzy ART unsupervised neural network was analyzed and developed. This network architecture is very good at novelty detection and can perform quick learning of new localized data without destroying the other stored information. The analysis of Fuzzy ART is presented in Chapter 4. The theoretical operation of the technology is described, including a modification to use individual case vigilance techniques, which allows the network to learn higher amplitude signals to a higher degree of precision than of low level noise. The input vector features are then developed, with background information describing the choice and composition of each feature. The network is then tested and the presentation of the information to the network is adjusted, based upon a priori information concerning spectral features to allow better separation of these features in the neural network representation of the spectrum. The resulting network correctly identified over 96% of the small changes that were applied to each of the components in the spectrum. As the size of the changes increased, the detection performance of the network increased.

Supervised Neural Network Performance Survey

Multiple neural network architectures were analyzed with the features developed in Chapter 2, both to check the utility of these features and to find the relative strengths of the various architectures. Chapter 4 presents an in-depth analysis of the Fuzzy ART architecture using the entire vibration spectrum, presented as frequency and amplitude, as the input features.

The following neural networks were built in NeuralWare Professional II Plus, and compared with respect to their ability to distinguish turbine operating modes, given the features developed in Chapter 2:

1. Multilayer Perceptron trained with the back-propagation learning rule
2. Probabilistic Neural Network
3. Learning Vector Quantization network
4. Radial Basis Function network
5. Fuzzy ARTMAP, the supervised form of Fuzzy ART
6. Modular Neural Network

Data Acquisition, Feature Extraction and Preprocessing.

A system encompassing data acquisition, preprocessing, and neural network operation was created for the supervised network analysis. A block diagram of the system is presented in

Figure 3-1. Following a survey of the LM2500 vibration data tapes, sections of the tape corresponding to different operating modes were digitized into separate files. The digitization system is described in Appendix A. The software for the digitizer consisted of two sets of C and assembly code, one set for the TMS320C31 DSP

processor, and one set for the PC host Pentium Pro processor. The DSP code controlled the four channels of digitization (two vibration channels and two tachometer channels). The PC interfaced to the DSP through dual-ported RAM to obtain the digitized signals. The PC code then assembled the raw data into a four-dimensional matrix in Matlab *.mat file format and stored the resulting signal on disk.

A set of Matlab m-files, or scripts, retrieved the individual signal files, performed calibration on the raw data, and extracted the time-domain and frequency-domain features. The feature set is shown in Table 3-1. The K-factor, crest-factor, and form-factor were included in the feature set, but not the peak, or peak-to-peak, average, or rms values, because the K-, crest-, and form-factors were linear combinations of the peak, average and rms values. The linear combinations provide a set of derived values that are important by themselves and do not provide redundant information when used without the peak, average, and rms values.

The set of seven features is assembled into a single record in the *.txt file, with each record being one input case to be preprocessed as input to the neural network. A case is single presentation of an input data vector to all the inputs of the neural network. It is also termed "example," "pattern," or "sample." The supervisory information is also included in each case.

There are three modes that the neural networks were trained to detect:

1. Turbine steady speed
2. Turbine accelerating
3. Turbine decelerating.

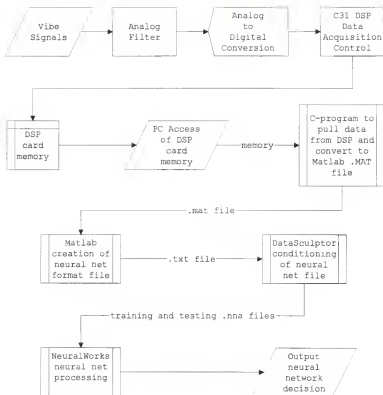


Figure 3-1. Supervised network analysis system.

In early tests of the neural networks without using the frequency data, especially combustion noise, the neural nets had difficulty detecting the acceleration mode. Because combustion noise is reasoned to increase as fuel is burned to accelerate the turbine, the peak level of the spectral component related to combustion noise was included in the feature set. The inclusion of combustion noise greatly improved the capability to resolve the acceleration mode.

Table 3-1. Features developed in Matlab program.

Feature	Description	Domain
1	Kurtosis	Time
2	Crest factor = peak / rms	Time
3	K-factor = peak * rms	Time
4	Form factor = rms / average	Time
5	Maximum combustion noise amplitude	Frequency
6	11-16th blade pass amplitude	Frequency
7	11-16th blade-pass high sideband amplitude	Frequency

Once the features were prepared and stored with the supervisory data on a case-by-case basis, the NeuralWare DataSculptor tool was used to perform normalization and to separate the data into training and testing files. DataSculptor allows the user to create a data transformation scheme that prepares raw data for use by the NeuralWorks Professional II/Plus neural network system. The scheme used to transform the features for the supervised neural networks is shown in Figure 3-2.

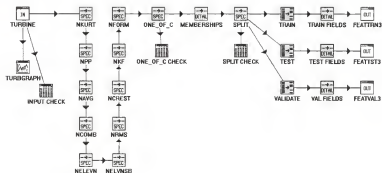


Figure 3-2. DataSculptor data transformation scheme.

Input data from the Matlab *.mat file enter the DataSculptor scheme through the TURBINE IN box. Attached to the TURBINE IN box are a graph (TURBGRAPH) and a tabular display (INPUT CHECK) that were used to ensure that the data sets were input correctly for the analysis.

Following the input box are a series of SPEC boxes that were used to normalize the various input feature vectors. The features, such as kurtosis in the NKURT box, were checked for their minimum and maximum extents, and normalized to have a minimum of -1 and a maximum of 1.

After the normalization SPEC boxes, a SPEC box labeled ONE_OF_C converts the supervisory data into a 1-of-C coding. The classification from Matlab contained one field, with a value of 1, 2, or 3, signifying stable operation, acceleration, or deceleration, respectively. The ONE_OF_C block creates three feature fields and populates them as shown in Table 3-2.

Table 3-2. One-of-C coding.

Input Supervisory Value	1-of-C Field 1	1-of-C Field 2	1-of-C Field 3
1	1	0	0
2	0	1	0
3	0	0	1

After the 1-of-C coding, the block called MEMBERSHIPS separates the data into training and testing datasets, using a random sampling of the original dataset. The SPLIT box separates data based upon the 1-of-C coding. The TRAIN, TEST, and VALIDATE sieve blocks fill datasets for training, testing, and validation files. The testing file performs the validation function of using the same dataset across different neural networks. Due to the scarcity of data, the validation set was

rarely used. To get each set of features, 1.5 MB of digital signal data had to be analyzed just to obtain seven data points. Accelerations happened very rapidly and there were relatively few of them on the recorded tapes. The low number of examples of acceleration was the limiting factor in obtaining the same number of acceleration, deceleration, and stable state cases in each dataset. Enough data sets were obtained, from each of 50 operating points, to perform the operating mode test and to compare the different neural networks.

The DETAIL boxes perform a straining operation on the available data in the datasets to ensure that just the required features and the 1-of-C coding enters the output files. The OUT boxes store the data on the disk for use in the neural network analysis. The same testing and training data sets were used in each of the architectures investigated below.

Neural Network Comparison

After the training and testing data sets were prepared and stored on disk, the NeuralWorks Professional/II Plus neural network prototyping and development system was used to analyze the signals. This system allows creation of over two dozen different types of networks and permits creation of new networks. Many tools are supplied to gauge the performance of the networks and monitor the training activity in progress.

The description of the back-propagation network is more detailed than the other supervised networks to provide an understanding of the network diagrams and performance measures. The survey primarily includes a short description and performance result of the network,

along with an excerpt from the NeuralWorks diagram of the trained network.

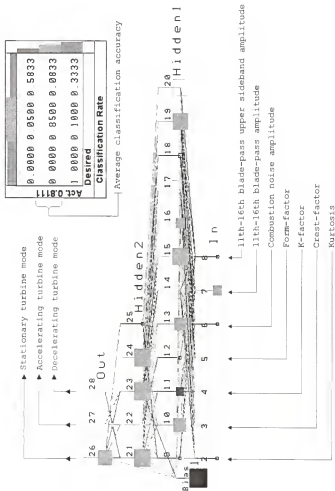
Back-propagation. Of the supervised neural networks, the multilayer perceptron, trained with the back-propagation learning method, is likely the most popular. This network began to be used in the mid-1980s, when the back-propagation learning scheme was developed [24]. Back-propagation trains by propagating the network errors back through the individual neurons based upon the gradient of change in error that is contributed by the individual neurons in the network in response to the input. A small change is made at each case presentation, to make the network slowly converge toward a minimum total error. Back-propagation has a problem finding the global minimum error point and can tend to become stuck at local minimums. Techniques such as momentum, noise-insertion, and weight-jogging have been invented to try to improve the convergence properties of back-propagation.

Back-propagation uses layers of neurons interconnected with weights. In the general case, where layer-skipping is not used, each neuron in a layer is connected to the outputs of the neurons in the preceding layer. Each neuron sums all the inputs, multiplied by the associated weights, to determine the activation of that neuron. Then an activation function creates an output value, based upon the internal activation. The activation function can be either linear or a differentiable nonlinear function. The activation function must be differentiable for the back-propagation function to work. In the back-propagation network presented in Figure 3-3, the input layer (labeled In) contains neurons with a linear activation function. The remaining layers use the hyperbolic tangent nonlinear activation function, that allows bipolar input to generate an output between -1 and 1.

The network presented in Figure 3-3 represents a back-propagation neural network with two hidden layers, an input layer, an output layer, and a bias node, which is always set to 1. The individual neurons, or nodes, on the network are shown as squares, with the size and shading of the squares indicative of the output values of each neuron. The outputs can assume values within the range of -1 to 1. A larger, darker square has a value close to 1 and a larger clear square has a value closer to -1. Smaller squares have values closer to zero. Weights associated with each interconnection are also shaded differently, depending on their magnitude, but the individual weights are hard to see on the printed diagram and are provided for only to indicate the network architecture. Figure 3-3 shows the input features being presented to the In layer and the output features emanating from the Out layer. All nodes are numbered sequentially.

Also shown on Figure 3-3 is a Classification Rate diagram. This diagram presents the percentages of correctly classified cases. There are three horizontal columns and three vertical rows. The columns contain the desired, supervisory expectations, and the rows contain the percentage of the case classifications. A perfect classification would consist of all ones on the diagonal that extends from the lower-left corner to the upper-right corner. Off-diagonal elements represent the percentage of errors in classification. The number shown on the left-most edge of the Classification Rate diagram is the average percentage of correct classifications.

Many different back-propagation neural networks were tried using the NeuralWorks tool. Different numbers of layers and different numbers of nodes in each layer were tested. This iteration of the network appeared to provide the best performance.



This network iteration required approximately 40,000 epochs to reduce the rms error below 1% in training and correctly classified 81% of the input cases. Better classification was sporadically obtained with less training. The stopping criterion for training was the reduction of the rms error below 1% error in classification of the training set. A better training criterion could be to periodically test the network against the training set, and stop when the error reaches a preset value. The best performance should also be recorded on an ongoing basis to retain information after extensive training that does not reach the preset goal. The term epoch refers to the cycling of a complete training set through the neural network during the learning phase.

Back-propagation networks work by creating hypersurfaces in the problem space, which separate the input cases into two regions. Each node, or perceptron, in the network provides one linear separation hypersurface. To use the back-propagation network to classify the peaks in the spectrum, (as was done using the Fuzzy ART network to be presented later), would likely require three or four nodes per classification in order to create a polygonal area around the peak of the feature. With 32768 input cases in the spectrum, there could be an exceedingly large amount of nodes (not to mention the length of the training time), while Fuzzy ART uses only 228 nodes to perform the classification.

Probabilistic Neural Network (PNN). This network performed slightly better than the back-propagation network, in that it learned 83.3% of the correct classifications in under 10,000 epochs. A diagram of the Probabilistic Neural Network is presented in Figure 3-4, where only 21 of the 50 nodes in the pattern layer are shown. The Classification Rate diagram shows good performance. The magnitudes of

the weights in the pattern layer are shown in the Prototype Weights diagram.

The PNN uses a neural network implementation of a Bayesian classifier [25]. The probability density function (PDF) of the input features is required to perform the classification. A Parzen estimator is used to create the PDF of the input features in the PNN. PNN systems generally required a large amount of data to develop adequate classifications. Although the classifications performed in this exercise were relatively accurate, it is expected that more input cases would produce a better estimate.

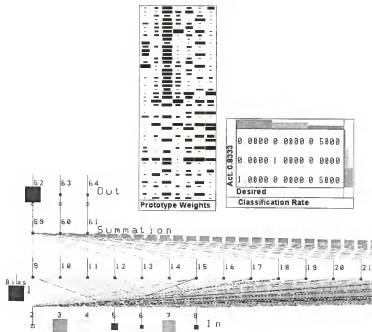


Figure 3-4. Probabilistic Neural Network.

Learning Vector Quantization (LVQ) network. LVQ networks contain a classification vector for each of the possible outputs of the network. The array of vectors is trained as a Kohonen network, which is a self-organizing feature map. As the input cases are presented, the vector with the output nearest to the correct output is adjusted to move slightly closer to the correct value, while a repulsion mechanism causes the other vectors to move away from this classification.

The performance of the LVQ network exceeded that of the back-propagation and PNN in this case, with a classification rate of 89.6%, only requiring 6000 epochs to reach this accuracy. A section of the network diagram is presented in Figure 3-5. It can be seen that each output is driven by a separate vector and that the input features are fully connected to each Kohonen layer node (nodes 9 through 59).

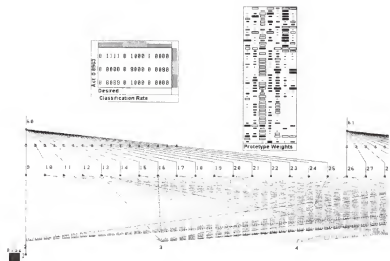


Figure 3-5. Learning Vector Quantization Network.

Radial Basis Function (RBF) neural network. The RBF network attempts to improve the separability of network classification by converting the input data into a higher-dimensionality representation. The pattern units of an RBF are characterized by a central point, from which the distance is measured in an Euclidean manner to determine the membership in the class represented by that neuron. RBFs were developed from interpolation theory and thus show good capabilities for function representation.

The RBF network showed the worst response to training on the dataset, after the General Regression Neural Network, which is not presented here due to its performance. After training for 10000 epochs, it correctly classified only 76.6% of the input patterns. More training epochs did not provide any improvement. The network performance could likely be improved with detailed analysis and tuning, but the main thrust of this research concerned the unsupervised Fuzzy ART network due to its ability to present a solution to the turbine monitoring problem without requiring defect response data. An excerpt of the RBF network is presented in Figure 3-6, where the higher-dimension pattern layer is seen as neurons 9 through 59.

Fuzzy Adaptive Resonance Theory, Mapping (Fuzzy ARTMAP) network. The unsupervised form of Fuzzy ARTMAP is presented in detail in Chapter 4, but the supervised form is appropriate for classification problems such as turbine mode detection. Fuzzy ARTMAP consists of two Fuzzy ART modules, connected by an inter-ART layer. The inputs are learned by the ARTa side and the outputs are learned by the ARTb side. The inter-ART module provides connections between ARTa and ARTb and forces searches of the ARTa side to occur to support the chosen output pattern mode of ARTb. The resonance searches and reset waves are discussed in the chapter on Fuzzy ART. Fuzzy ARTMAP is used here to classify the

turbine operating mode, but it was also used in the initial investigation into the spectral analysis system described in Chapter 4. The difficulty with the spectral analysis application was the lack of defect response data for training. The Fuzzy ARTMAP network would likely been able to detect patterns in the spectrum indicative of known turbine defects. The input data to this Fuzzy ARTMAP application would be preprocessed identically to that used for the Fuzzy ART application presented in Chapter 4, but would include supervisory data to flag known defect responses.

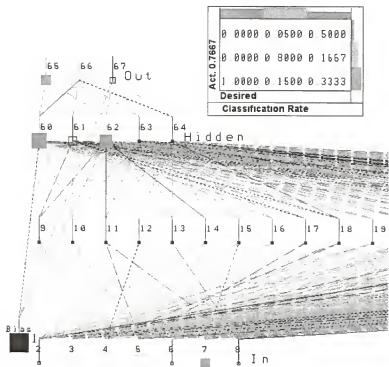


Figure 3-6. Radial Basis Function Network.

The implementation of Fuzzy ARTMAP in NeuralWorks, shown in Figure 3-7, provides just one Fuzzy ART module, instead of the two normally used. They compensate by forcing a resonance search based upon mismatches in the output layer. This technique should be an acceptable compromise, but enforces constraints on the type of data presented to the output layer. This simplification was done to permit their Fuzzy ARTMAP to perform 1-of-C classifications. In a true Fuzzy ARTMAP implementation, as was hand-coded in C for experiments in this research, the output Fuzzy ARTMAP module (ARTb) provides a much richer capability to generate arbitrary output responses to input cases. Instead of a single output layer, there is a set of neurons, with each neuron providing a different learned response.

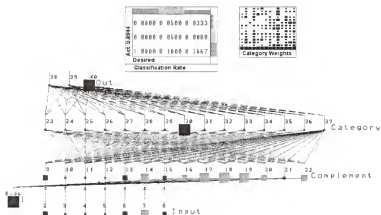


Figure 3-7. Fuzzy ARTMAP Network.

NeuralWorks combines the F_1 and F_2 layers (as described in Chapter 4) into a single Category layer and show the F_0 preprocessing layer as the Complement layer. The need for the bias node is unclear

because there is no bias node needed in standard Fuzzy ARTMAP equations. The performance of the Fuzzy ARTMAP in this implementation provided good classification, second only to the Modular Neural Network. It yielded an accuracy of 89.4%, required approximately 10 epochs to train, and used 14 category nodes.

Modular Neural Network. This may be one of the most interesting neural network architectures currently being investigated. It did achieve the highest average classification rate, 95.0% correct classifications, higher than the next best, Fuzzy ARTMAP. The major distinguishing feature of this network is the use of multiple expert network modules that learn different facets of the problem. A gating mechanism switches in the appropriate expert to supply the best-learned output for a given input case. The network diagram of Figure 3-8 shows the gating network and one of the hidden expert subnetworks (called local experts [17]).

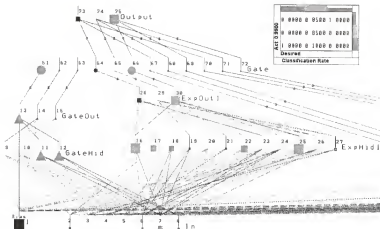


Figure 3-8. Modular Neural Network.

In the Modular Neural Network diagram, there are two more local experts off-page. The gate switches the outputs of one or another local experts to the final output. The input layer is fully-connected to each of the local experts and the gating network. The gating network and the local experts are trained with the back-propagation learning technique. The gating network permits a form of competitive-learning between the local experts.

CHAPTER 4

FUZZY ADAPTIVE RESONANCE THEORY NEURAL NETWORK APPLICATION

Gail Carpenter and Stephen Grossberg created the Adaptive Resonance Theory (ART) neural network technology [26] [27] at Boston University in the late 1970s. Its capabilities have continued to be enhanced with new forms of the architecture, including the ability to handle analog/fuzzy inputs instead of strictly binary inputs, and to function in a supervised manner. With ART, Grossberg achieved much success in creating a network that met his long-standing goal of creating a stable learning technique that would not destroy previously learned information when it learns new information.

The technology was created in an attempt to imitate cognitive and neural processes that are believed to occur in the brain. It implements a type of short-term/long-term memory architecture that stores accumulated knowledge in prototype vectors, imitating long-term memory storage in the brain. A short-term memory subsystem investigates new information presented to the network. The network checks to see if the new information corresponds, or resonates, with information stored in long-term memory. If a new input vector is presented that had not previously been learned, the system can learn it immediately, without destroying its other knowledge, and can recognize the input vector as novel.

ART is thought to imitate higher-level thought processes within the brain in that it allows the network to respond to input cases while viewing the case through all its stored knowledge in a manner similar

to human perception. A subsection of the ART architecture called the attentional subsystem weighs the new information with previously learned information to look for a suitable match, using a competitive process between neurons to determine which neuron best represents the input vector. The concept of resonance allows neurons to be checked for a predetermined quality of match to the input information. There are feedback mechanisms in the read-out of the stored information that provide stability to prevent the network from ceaselessly changing its weights in response to new inputs. An orienting subsystem, or novelty detector, allows sufficiently different input vectors to cause localized changes in the stored weights. Thus, ART exhibits very good performance in the "stability-plasticity dilemma," where it is stable from extensive continuous change, but is plastic, or changeable, when sufficiently novel information is presented.

Engineering Description

Fuzzy Adaptive Resonance Theory (Fuzzy ART), as other neural networks, creates internal templates, or weight vectors, that represent the vector data presented to the system. Each neuron contains one template. Each neuron is initialized with data from a single vector. It is then said to resonate with that vector. As new vectors are presented to the network, the proximity of the new vector to the weight vector of each neuron is compared. Closer proximity results in higher resonance. The resonance level is thresholded against a system parameter called vigilance. If the resonance is above the threshold, then the neuron resonates with the new input vector. The new input vector can then be learned by the neuron. The neuron will no longer represent a single vector point, but will now represent a subset of the

input vector space. The subset of the vector space represented by a neuron can only grow to a maximum size, governed by the vigilance parameter. The central location in vector space represented by a single neuron is not capable of changing throughout the vector space, but must remain within a maximum distance from the vector originally programmed into the neuron's weights. The ability to resonate with a constrained, stable subset of the input space provides essential functionality for long-term detection of changes in features of input vectors presented to the network.

Fuzzy Adaptive Resonance Theory (Fuzzy ART) networks have evolved from earlier networks that employed associative learning rules, such as Hebbian, Instar, and Outstar, to classify or represent vectors of input information. Neurons exhibiting these learning rules were grouped into neural networks to perform useful functions, such as in the self-organizing map (SOM) network. In the SOM, the weights of the neurons become representations of the input vector. Starting from an initial condition, the weights are adjusted to represent a set of input vectors.

Fuzzy ART uses some of the principles of these earlier networks, but it adds functionality in order to partition the input space in a more deterministic manner. This functionality includes the use of the vigilance test to parametrically test a weight vector against an input vector, and the representation of ranges of vectors in the weights instead of single vectors. Elements of earlier neural networks, including Hebbian, Instar, and Outstar learning, have influenced development of Fuzzy ART. The earlier networks are discussed and compared with Fuzzy ART in the following sections.

Hebbian learning. In Hebbian learning, the weights of the neurons are adjusted to learn binary input vectors. Starting with a

set of neurons that had orthogonal initial weights, input vectors are compared against the individual neurons. If a vector element matches an element in the neuron weight vector, that neuron's weight is updated by increasing the magnitude of each weight corresponding to an element in the input vector.

Fuzzy ART uses analog inputs instead of binary inputs, but will still update the weight vector of the neuron using information from every element of the input weight vector. The original ART vector performed in a more Hebbian manner because it used binary inputs. New input information would cause the elements of the weight vector to match the elements in the input vector. ART limited the amount that a weight vector could be modified using the resonance parameter. Fuzzy ART will also detect if the input vector is represented by the weights of a neuron, and learn the new information contained in the input vector, in a manner similar to Hebbian learning. Fuzzy ART can respond to both positive and negative going changes in the input vector, whereas Hebbian monotonically increases the neuron weight vector as information is learned.

Instar learning. The instar learning technique was invented by Stephen Grossberg, to try to control the stability of information learned by a neural network. The Instar neuron performs a vector recognition function. Instar uses fuzzy or analog inputs as opposed to the discrete inputs of Hebbian learning. Instar learning provides a competitive learning technique, where only the neuron that exhibits the highest activation in response to the input vector is allowed to update its weights. Each neuron contains a set of weights of the same dimensionality as the input vector. When an external stimulus is applied (such as the results of an activation calculation), the weights are allowed to slowly adjust toward the values contained in the input

vector. After training, the Instar neuron will provide an output that indicates how close the input vector is to the prototype vector that it has learned [30]. Thus, the Instar neuron performs a vector detector function.

Instar learning has some features in common with the Fuzzy ART neuron, in that

1. It uses analog (or fuzzy) inputs.
2. It learns only when gated by an external stimulus. Thus, learning is localized to a single neuron and not the entire network. In Fuzzy ART, the weights of only one neuron at a time are modified in response to any input. A neuron will not learn the input vector until gated by an external stimulus. The stimulus that gates the weight modification is the activation and resonance checking mechanism.
3. A competitive learning mechanism is provided that checks the activation of the neuron with respect to the input vector.

Unlike the instar neuron, Fuzzy ART neurons cannot change their weights throughout the input space after being initially committed. Their weights can only be adjusted within a small range, governed by the vigilance parameter. After winning the activation competition in Fuzzy ART, the node must pass a resonance test that checks the closeness of the winning node to the input vector. A good winner will have strong resonance, and will pass the resonance check. A bad winner will not pass the resonance check, and will not be used to represent the input vector. A new search of the activations for the next winner will ensue. If no neurons provide a good winner, a new neuron will be committed to represent the novel input vector. Thus, Fuzzy ART limits the amount of change performed by any single neuron.

Outstar learning. Outstar neurons perform an opposite function as Instar neurons. They generate a vector in response to a stimulus. This Instar neuron generates an output in response to a vector input that is close to the vector that it had been trained to recognize. The Instar and Outstar neurons can be combined into a larger network, for instance to regenerate a vector input that had been corrupted with noise. The Instar neuron responds to the noisy vector as closely matching the prototype vector triggering a corresponding Outstar neuron to output the prototype vector.

The vector generation capabilities of the Outstar network are not used in Fuzzy ART, but the supervised form, Fuzzy ARTMAP, has such a correspondence. In Fuzzy ARTMAP, there are two interconnected Fuzzy ART sections. The input section is trained to distinguish the input vectors and the output Fuzzy ART section is simultaneously trained to reproduce an appropriate output vector. The input training patterns are learned by the input Fuzzy ART layer (ARTa) and the corresponding output patterns are learned by the output Fuzzy ART layer (ARTb). The Inter-ART layer provides a tracking function that causes the input layer to reset initial resonant choices if they do not result in correct predictions and to learn new ARTa categories, if required, to cause the correct output category to be selected. When a match is made, the output layer provides a learned output vector. In both the Outstar and the Fuzzy ARTMAP network, only one output neuron is activated at a time.

Operation of Fuzzy ART. The ART neural network extended neural network architectures, particularly the Instar, to introduce a resonance capability. New vectors of input information presented to the network would be checked for a sufficient match with previously learned vectors through the resonance concept. Vectors not causing

sufficient resonance would be recognized as novel and learned without causing extensive modification to previously learned information. The concept of vigilance is central to the ART technology. The vigilance parameter controls the amount of resonance that the network requires before classifying an input vector as being a sufficient match to previously learned information.

Another major change in the operation of Fuzzy ART versus the historical networks is that Fuzzy ART learns a range of vectors in each neuron. The vigilance parameter controls the size of the subset of the input range, or fuzzy subset [28] that may be learned by a single neuron. The weights in a Fuzzy ART neuron store information describing the extents of the fuzzy subset represented by the neuron. This fuzzy subset is completely defined by two points in the input space. Therefore, the weight vector in a Fuzzy ART neuron has a dimensionality twice the size of the input vector, holding the extreme points of the fuzzy subset.

Figure 4-1 illustrates the concept of resonance and the extremes of the fuzzy subset stored in the weights of the neuron. Three vectors have been learned by a single neuron. The neuron prototype has expanded to a range that encompasses all three vectors. The weight vector in the Fuzzy ART neuron describes the extreme points U and V, of the neuron's prototype, or fuzzy subset of the input range that describes the vectors that have been learned.

The vectors are shown normalized to a range of [0,1]. The prototype is shown with a resonance field surrounding it. The resonance field describes the maximum expansion size for the current prototype. As the prototype learns new vectors, the prototype and associated resonance field size will change. After the prototype has reached the maximum size, governed by the vigilance parameter, there is

no resonance field, and the neuron can learn no more new vector information.

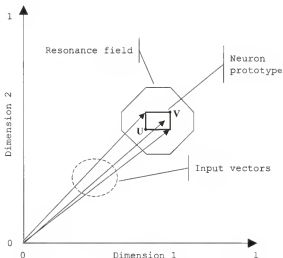


Figure 4-1. Vector illustration of Fuzzy ART.

When presented with the first input vector, Fuzzy ART initially commits a new neuron to represent the input vector. The neural weights are set to represent the dimensional lengths of the vector. As new vectors are presented to the network, and these vectors are sufficiently close to the stored prototype, the Fuzzy ART neuron adjusts its prototype weights to encompass the range of values of the new vectors. If a new vector is presented to the neuron, and falls within the resonance field of that neuron, the neuron will learn the vector and expand its prototype to encompass the new vector, as shown in Figure 4-2.

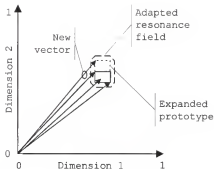


Figure 4-2. New vector learned by Fuzzy ART.

If new vectors were presented that fall outside the resonance field of a certain neuron, then that neuron will not expand its prototype to learn the new information. This is shown in Figure 4-3.

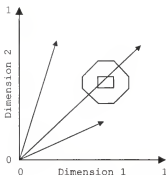


Figure 4-3. Vectors not resonating with neuron prototype.

In Fuzzy ART, a vigilance parameter governs the maximum expansion size of a prototype region. Training with a high vigilance value causes Fuzzy ART to generate small prototypes, thus making classifications that are more precise, but requiring more neurons to

perform the classification. A lower vigilance value commits fewer neurons, but creates a rougher classification, with larger prototypes.

The original ART network, ART1, was developed with intrinsic nonlinear differential equations and handled binary cases exclusively. The Fuzzy ART architecture expanded the network to allow processing of analog information in the input vector. Fuzzy ART uses the fuzzy-min and fuzzy-max set grouping operations from fuzzy logic to perform the input and analysis functions. Binary values and general analog values, without fuzzy connotations, can also be used as input. The results are interpreted in the universe of discourse relating to the subject matter.

In classical fuzzy logic systems, the truth of a proposition is represented as a continuous function instead of as a binary answer, such as true or false. For instance, the question "Is it a sunny day?" could be answered with a yes or no response, but a response like "It is slightly sunny." or "It is very sunny." are still valid responses in ordinary speech. Fuzzy logic provides a method to manipulate this type of inexact, or multivalent, answer mathematically [28], [29]. The truth of the statement would be ascribed to a range from zero to one. In reference to the "sunny day" example, a response of 0.2 may represent a highly overcast, cloudy day, but a response of 0.9 may represent a very sunny day. While Fuzzy ART allows interpretation of the inputs in term of fuzzy logic, it is equally valid to apply no fuzzy connotations to the input. Fuzzy ART has the ability to process analog information, in the range of zero to one. It was used in the research in this manner. The input consisted of amplitude levels and frequency from the turbine vibration spectrum, along with a priori knowledge about the spectrum that was coded into an analog representation.

Fuzzy ART has some characteristics that make it well suited for online vibration signal monitoring. These characteristics include the following:

1. High plasticity and stability. As new spectral information is learned, old information is not lost. If categories need to be changed to accommodate new information, only the single category that best resonates with the new input vector is altered. If no categories resonate sufficiently, a new neuron is dedicated to hold the new information. This is similar to the operation of the Upstart algorithm [31] that adds nodes as needed to lower the output error. In contrast, learning new information using a back-propagation algorithm has the potential to modify the weights of every node at each case presentation; thus causing unselective forgetting of information previously learned [32].
2. Quick-Learning. A new input vector can be learned as it is presented. The entire training set does not have to be cycled through to learn a single new vector of information. Fuzzy ARTMAP, the supervised form of Fuzzy ART, has been documented to learn the two-spiral benchmark in five epochs while it takes up to 20,000 epochs for back-propagation [32]. In the turbine spectrum monitoring application, Fuzzy ART learned a 65536-point input spectrum in two epochs, requiring less than one second on a personal computer.
3. Novelty-detection. Fuzzy ART detects input vectors that do not sufficiently resonate with previously learned information. This forms the basis of its application in vibration analysis. As the spectrum changes, the neural network can detect the new spectral characteristics. The spectral change information can then be recorded and immediately learned as a new baseline.

4. The Fuzzy ARTMAP prototypes directly represent the learned information, easing interpretation of the operation of the network and the reasons behind its decisions. In contrast, it is generally difficult to interpret how the weights of networks such as back-propagation produce their output given the input vectors.
5. No Local-Minima problems, by not using the gradient-descent weight-modification method of back-propagation nor error-surface minimization searches in training.

In the application of Fuzzy ART to spectrum comparison, a detailed spectrum of gas turbine vibration was used as input to the neural network. Over 200 neurons were required to represent the complex spectrum. Each input vector contained a single frequency component and the associated amplitude of the spectrum at that frequency. One presentation of the spectrum to the network consisted of cycling through all the amplitude/frequency components in the spectrum. The frequency dimension was linearly increasing, while the amplitude dimension assumed various values within the normalized range of $[0,1]$ as determined from the spectrum. At the completion of training, Fuzzy ART had created a representation of the spectrum, accurate to the level of vigilance used in training. Changes in the spectrum could then be detected using the novelty detection capabilities of the network.

The 20kHz spectrum bandwidth was represented with 65536 frequency components. Fuzzy ART requires an input range of $[0,1]$, so the spectrum was normalized, or standardized, to represent the full 20kHz bandwidth within the range $[0,1]$. The amplitude range was also normalized to represent 49 dB of amplitude within the range $[0,1]$. The frequency and amplitude components were thus spaced closely together in

the normalized representation. Careful control of the Fuzzy ART parameters was required in order to ensure that the network was trained successfully and yielded useful results. The network was trained to detect an amplitude change of 5% of the full range [0,1]. The detection of a 5% change in amplitude required that the maximum resonance field be less than 5% of the amplitude dimension. The resonance field extends equally through all input dimensions, also encompassing 5% of the frequency dimension range. The spectral components of the gas turbine were spaced together closer than 5% of the frequency range; thus yielding false positive classifications as changed spectral components resonated with components nearby in frequency. A method was developed to separate the frequency components using additional neural inputs.

The frequency dimension was a linearly increasing metric that spaced the amplitude components at discrete points throughout the input space. The spacing was too tightly grouped for effective separation of some information in the spectrum, so more separation was provided by augmenting the input vector with additional input dimensions. The frequency range was segmented into sections and the additional dimensions were then changed slightly for each corresponding frequency section. The dimensional change was calculated to sufficiently offset the input in vector space, thus separating adjoining resonance fields.

Two methods of segmenting the frequency axis were used in combination. Both methods relied on a priori information about the gas turbine spectrum. One method used the smallest expected separation of known frequency components in the spectrum to determine the section length. The section length was chosen to be smaller than the spectrum component separation. The other method used the frequencies of known turbine features to center a segment on each feature. A priority

scheme was used to ensure that a separate contiguous segment was established for each important feature in the spectrum. The effect of the offset on the input vector space can be seen in Figure 4-4, where sections of the spectrum are shifted to different location in a new dimension. The sections of spectrum appearing near zero in the A Priori Coding dimension are due to the slicing of the spectrum into equal parts. The dimensional offset is small in this area, sufficient to shift the sections out of possible resonance with each other. There are other sections of spectrum exhibiting a greater offset in the A Priori Coding dimension. These sections correspond to the higher priority segments corresponding to known features in the turbine spectrum.

There were 819 different a priori classifications used to segment the frequency dimension. Each segment had to be separated by an ℓ^1 -norm distance sufficient to prevent the extension of resonance fields from one segment into the next. This was accomplished by adding three additional components to the input vector. A small but sufficient offset was arithmetically added to the set of components for each of the 819 a priori classifications. When one component reached the maximum value of 1, it was reset and an offset was added to the next component, in an arithmetical carry operation. The frequency segments could have also been separated with 1-of-C coding, where 819 new dimensions would have been included in the input vector, but the network size would have grown unmanageable. It was unnecessary to use 1-of-C coding when addition of small offsets was all that was needed to ensure separation. These small offsets amounted to only a fraction of the maximum range, $[0,1]$, of each additional component. This technique was termed fractional-dimension 1-of-C coding. It represented 819

different segments using only three new input vector dimensions. Because the three additional dimensions could not be shown in Figure 4-4, the 819 classifications were shown as occupying just one additional dimension for graphing.

The operation of Fuzzy ART and the development of the spectrum monitoring application will be discussed in detail in the sections to follow.

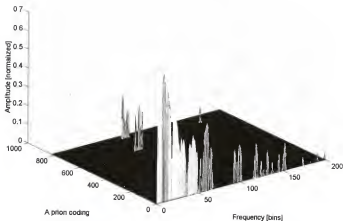


Figure 4-4. Effect of a priori coding on input vector space.

Fuzzy ART Theory

A Fuzzy ART system consists of two major components, the attentional subsystem and the orienting subsystem. The attentional subsystem activates the system in response to the input vector and the

orienting subsystem finds the correct internal representation of the new information.

Fuzzy ART creates two internal representations of vectors presented to the system. There is a long-term memory (LTM) layer that represents information that the system has learned and a short-term memory (STM) layer that perceives and internalizes new cases presented to the system. The long-term memory is stored in neurons, that are also called processing elements (PEs), or nodes, contained in the F_2 layer. The short-term memory is formed by neurons in the F_1 layer. The long-term memory neurons hold categories or templates that consist of hyper-cubic areas that encompass the region of the input space representing the category of information that it has learned. The short-term memory tests the input vectors for degree of match between the long-term memory and the input vector.

Figure 4-5 shows the different Fuzzy ART layers, the short-term memory F_1 layer, long-term memory F_2 layer, the attentional subsystem, and the orienting subsystem. The F_0 layer shown in Figure 4-5 provides preprocessing of the input vector to allow both low-amplitude and high-amplitude information to affect the network in the same manner. The input layer and the F_0 layer both consist of a single PE containing a vector representation of the input case. The F_1 and F_2 layers each contains a set of PEs. The F_2 layer contains the same number of PEs as the F_1 layer. Each PE in the F_1 and F_2 layers contains a vector of weights of the same dimensionality as the processed input vector $\mathbf{I}$ in the F_0 layer. The number of weights in $\mathbf{I}$ is twice the number of dimensions in the input vector $\mathbf{a}$.

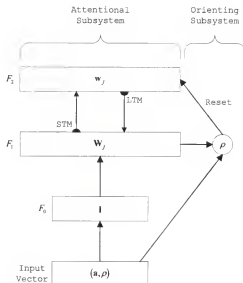


Figure 4-5. Fuzzy ART System Diagram.

The top-down weight vectors in the F_2 layer are called w_j , with the j subscript signifying the neuron index number in the F_2 layer. The bottom-up weight vectors in the F_1 layer are called W_j , with the j subscript signifying the PE index number in the F_1 layer. For each F_2 node, w_j , there is a corresponding F_1 node, W_j . The W_j weights are the short-term memory traces, (STM) and the w_j weights are the long-term memory traces, (LTM).

Figure 4-5 includes a variable called vigilance and denoted by ρ . The vigilance controls how close the system pays attention to the input vector. Generally, the vigilance is set to a constant for the whole training set. This makes all patterns equal in regards to how closely

the system learns the individual vectors. By using a priori knowledge about the distribution of the input information, the amount of vigilance that each vector gets can be set on a per-case basis before input into the neural network. Other terms for individual case vigilance are indivigilance or case vigilance. These terms will be used interchangeably. The case vigilance for input to the neural net can be set by hand, or modulated algorithmically, based upon features from the input case. Vigilance is formally described in the sections concerning the orienting subsystems and dynamic operation.

The description of Fuzzy ART requires both a network structure and a dynamic operation description to show how the network functions. The network structure description discusses the network neural interconnections, the weights, and the mathematics governing the changes to the weights. The dynamic operation description discusses the operation during learning and recall.

Fuzzy Set Operators

Two operators from fuzzy set theory are used extensively in Fuzzy ART. These are the fuzzy-min, denoted by the $\wedge$ operator, and the fuzzy-max, denoted by the $\vee$ operator [29]. The fuzzy-min is analogous to the set theory intersection, $\cap$, and serves as a conjunction of fuzzy membership in sets. The fuzzy-max performs a disjunction of the membership in the fuzzy sets. The operations performed are

$$p \wedge q = \min(p, q) \quad (5)$$

and

$$p \vee q = \max(p, q) \quad (6)$$

For example,

$$0.2 \wedge 0.6 = 0.2$$

and

$$0.2 \vee 0.6 = 0.6.$$

For a pair of two-dimensional vectors, the fuzzy-min or fuzzy-max of each component, or dimension, in the vectors is performed, with the result being a third vector. This is shown in Figure 4-6, where the fuzzy-min of vectors **A** and **B** results in the third vector shown as $A \wedge B$.

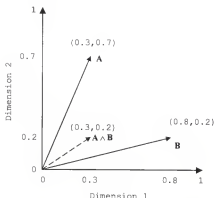


Figure 4-6. Fuzzy-min of two vectors.

The fuzzy-max of the same two vectors is shown in Figure 4-7. In the fuzzy-min, it is seen that the resulting vector assumes the minimum of each of the vector components. In the fuzzy-max, the resulting vector components assume the maximum of the corresponding components in the two input vectors.

Input Layer Description

Each vector presented to the network enters through the Input Layer. The input vector is labeled **a** and has a dimensionality **M**. The **M** different values in the input case vector **a** are labeled

$$\mathbf{a} = (a_1, a_2, \dots, a_i, \dots, a_M)$$

where the range of each a_i is

$$a_i \in [0,1] \quad (8)$$

Range notation in this document uses $[0,1]$ to show a range from 0 to 1, including 0 and 1. The notation $[0,1)$ indicates a range from 0 to 1 including 0, but not including 1. The notation $(0,1]$ indicates that the range from 0 to 1 includes 1, but not 0.

Any valid analog number, scaled between 0 and 1, or any binary number, having values 0 or 1, is acceptable as a value in the input vector. In the case of purely binary values being used in all cases, the weight arithmetic operations of Fuzzy ART reduces to that of the earlier ART1 network [26].

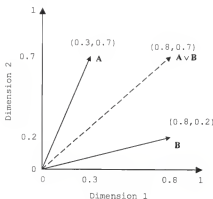


Figure 4-7. Fuzzy-max of two vectors.

Preprocessing Layer Description

The F_0 input layer preprocesses the input layer vector, $\mathbf{a}$, to extend the representation of the input vector to allow the network to

represent ranges of input vectors in a single neuron, as opposed to storing just a single vector. This is accomplished by doubling the dimensionality of the input vector to store a lowest-most corner and an uppermost corner of the range of vectors represented by each node in the network. The technique of complement-coding allows this range to be processed using the fuzzy-min operator.

The creation of the classification regions in Fuzzy ART primarily causes the individual values of the neural weights to monotonically descend towards zero as learning ensues, due the use of the fuzzy-min operator. This results in difficulty adjusting the classification regions to encompass information that may be increasing. Unless complement-coding is used, a great proliferation of categories can result as categories are continuously modified toward lower-valued case information [33] and new neurons must be committed to handle the new higher-value inputs. Complement coding encodes high-valued information as low-valued information to allow the hyper-cubic categories formed during learning to grow both more negative and more positive with the application of new input vectors.

Complement coding doubles the size of the input vector by creating the complement of each value in the input vector. The complement-coded representation of an input value a_i is

$$a_i^c = 1 - a_i \quad (9)$$

which is also of the range

$$a_i^c \in [0,1]. \quad (10)$$

Complement-coding the entire input vector $\mathbf{a}$ results in the pattern $\mathbf{I}$ that is formed in the F_0 layer, where $\mathbf{I}$ consists of the input vector values and the complement of the input vector values. For the following M -dimensional input case:

$$\mathbf{a} = (a_1, a_2, \dots, a_i, \dots, a_M),$$

the resulting complement coded input pattern $\mathbf{I}$ is a $2M$ -dimensional input vector:

$$\begin{aligned} \mathbf{I} &= (\mathbf{a}, \mathbf{a}^c) \\ &= (a_1, a_2, \dots, a_i, \dots, a_M, a_1^c, a_2^c, \dots, a_i^c, \dots, a_M^c) \\ &= (a_1, a_2, \dots, a_i, \dots, a_M, 1-a_1, 1-a_2, \dots, 1-a_i, \dots, 1-a_M) \end{aligned} \quad (11)$$

Thus, for an input case $\mathbf{a} = (0.2, 0.7)$, the complement-coded input $\mathbf{I}$ is $\mathbf{I} = (0.2, 0.7, 0.8, 0.3)$. The low-level $a_1 = 0.2$ value is complement-coded to a higher-level $a_1^c = 1 - a_1 = 0.8$ value. The F_0 layer holds the complement coded input vector. A vector representation is shown in Figure 4-8.

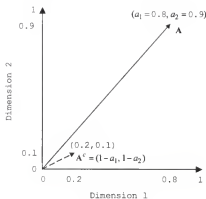


Figure 4-8. Complement-coding of vector $\mathbf{A}$, yielding $\mathbf{A}^c$.

Activity Layer and Category Layer Descriptions

The F_1 activity layer determines the amount of activity present in the different nodes when an input vector is presented. In the F_1 layer, the input vector is compared with the stored prototype information, using the fuzzy-min, to determine how close the new vector

is to the stored prototype. The F_1 layer is conceived to hold short-term memory traces to determine directions for further hypothesis testing to find a resonating F_2 category (or to learn a new F_2 category).

The F_2 category layer holds the prototypes that are checked for resonance with the input vector. This resonance check ensures that the network has found the right category to represent the new input vector.

The long-term memory traces in the F_2 layer are stored in neurons each containing a weight vector, or prototype, that represents the category. The input space is multidimensional, with one dimension for each value in the input vector and one dimension for the complement-coded representation of each value in the input vector. Each dimension has maximum extents of $[0,1]$, with the information coded as a subset of this extent. The information in a single-dimension vector forms a line segment within the $[0,1]$ range. A two dimensional template contains a rectangle that separates that category from the rest of the space. Three-dimensional templates form cubic regions. Dimensions higher than three form hyper-cubic templates in long-term memory that express the knowledge as bounded hyper-volume sections of the possible input space.

The weight vector, or prototype, stored in a single node of the F_2 layer is built up from the information from all the vectors that have resonated with the node. When a vector is first learned by the node, the prototype assumes the form of a single point, as shown in Figure 4-9. The general form of a two-dimensional prototype is that of a rectangle, with the lower-left and the upper-right vertices stored in the prototype, using complement-coding to store the upper-right vertex. With just a single vector learned, the upper-right and the lower-left

vertices are equal and the prototype represents just a single point in the input space.

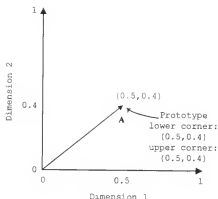


Figure 4-9. Single-point prototype, one vector learned.

When another vector resonates with the node, the information in that vector is also learned and the prototype is modified. An example of this is shown in Figure 4-10, where a second vector modifies the single point vector of Figure 4-9. The resulting prototype then assumes the form of a rectangle.

Because Fuzzy ART learns a range of input vector space instead of a single vector in each neuron, the activation of a neuron cannot be determined by a simple projection of the input vector onto a weight vector. The activation is determined by a form of projection of the input vector onto the prototype, using the fuzzy-min and fuzzy-max operations.

If a third vector resonates with the node, the prototype is again modified. The modifications of the prototype are performed using the fuzzy-min operation of the vector with the lower-left corner and the

fuzzy-max of the vector with the upper-right corner. This is illustrated in Figure 4-11, where a third vector is learned.

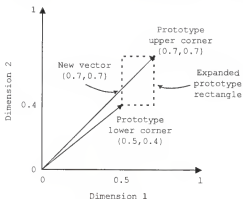


Figure 4-10. Rectangular prototype, two vectors learned.

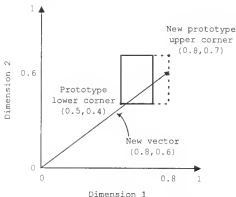


Figure 4-11. Expanded prototype, third vector learned.

The weights in the F_2 layer are expressed as

$$w_j = (w_{j,1}, \dots, w_{j,k}, \dots, w_{j,2M}) \quad (12)$$

of dimensionality $2M$ to hold both the lower corner vertex and the upper

corner vertex. The j index refers to a neuron in the F_2 layer. Each neuron in the F_1 layer contains the weight vector

$$\mathbf{W}_j = (W_{j,1}, \dots, W_{j,K}, \dots, W_{j,2M}), \quad (13)$$

with the index j representing the neuron number in the F_1 layer. The F_2 weights are termed the top-down weights and the F_1 weights are called the bottom-up weights. The bottom-up and top-down weights all have values that are in the range

$$w_{j,k} \in [0,1] \quad (14)$$

and

$$W_{j,k} \in [0,1]. \quad (15)$$

The F_1 and F_2 layers both have the same number of neurons. Each F_1 neuron is linked to the corresponding F_2 neuron.

The F_1 layer accepts the input $\mathbf{I}$ and compares it with a read-out from long-term memory, F_2 , to generate the bottom-up weights $\mathbf{W}_j$. The bottom-up weights are then analyzed in the F_1 layer to determine each neuron's activation $T_j(\mathbf{I}^P)$ in response to the new input $\mathbf{I}^P$, where the P superscript denotes the position of the input vector in the neural network input training set. The neuron that has the highest activation $T_j(\mathbf{I}^P)$ in response to the new input is then analyzed in the F_2 layer to determine if it has sufficient resonance to classify the input vector.

The $\mathbf{W}_j$ weights exist internally to the F_1 layer and do not propagate to the F_2 layer. The $\mathbf{W}_j$ weights are used to calculate an activation quantity that allows competitive selection of nodes for resonance testing. They are used to build up an internal representation of the compare the $\mathbf{I}$ vector with the top-down weights,

w_j . If the incoming vector has enough features alike with the prototype stored in long-term memory, then the F_1 layer activates the F_2 layer to take a closer look at the case, with respect to the vigilance setting. If a close enough match is found, then resonance occurs, and the long-term memory weights are adjusted, if needed, to encompass the new information in the input vector. If the resonance test fails to recognize the input vector, then the vector is deemed novel and a new F_1 and F_2 neuron pair is dedicated to hold the information contained in the vector. The result of recognizing a novel vector is available as a trigger for further extra-neural processing. This ability to reliably recognize novel vectors is crucial to the spectral analysis application. Novel input vectors, when viewed as changes in spectral features, can trigger a trending recorder to keep track of the changes in the spectrum, possibly indicative of machinery degradation.

Before programming, there is one neuron, or node, in each the F_1 and F_2 layers. This node is called an uncommitted node. After a pattern is learned by a node in the F_2 layer, that node is called committed. The number of nodes in the F_1 and F_2 layer grows as nodes become committed. The number of committed nodes is termed Maximum Committed Node, MCN . The system always keeps one uncommitted node available to support the learning of novel information. As the system learns, and nodes are committed, the number of nodes in the F_1 and F_2 layer thus grows to hold $MCN+1$ nodes.

The initial node values for the F_2 layer are designated $w_{j,k}(0)$ and its weight values are chosen as

$$w_{j,k}(0) = 1, \quad 1 \leq k \leq 2M. \quad (16)$$

Thus, the uncommitted node has all its weights set to 1. All uncommitted nodes have this property.

The F_1 layer weights are generated through calculations in the F_1 layer and are not set to an initial value, but an F_1 layer activation parameter α_j is initialized. The α_j parameter provides an initial ordering of uncommitted neurons, such that the neuron with the lowest index is committed first. This has prime importance when the Fuzzy ART network is realized in hardware, where a full layer of neurons would be constructed before programming.

The activation parameter satisfies the following inequalities:

$$\alpha_1 > \alpha_2 > \dots > \alpha_{MCN+1} \quad (17)$$

and

$$0 < \alpha_j < \frac{1}{\beta + M_u} \quad (18)$$

where the choice parameter, β , is used to indicate to the network how deep to search for a resonating category before committing a new node. The choice parameter satisfies

$$\beta > 0. \quad (19)$$

A frequently used value for the choice parameter β is 0.01.

The set-up parameter, M_u , is chosen to satisfy

$$M_u \geq 2M \quad (20)$$

where M is the number of values in the input vector $\mathbf{a}$.

The combination of values for β and M_0 allow the activations resulting from uncommitted nodes to be of similar magnitude to those of committed nodes.

After a node has been committed, the F_1 layer activation changes to

$$T_j(\mathbf{I}^P) = \frac{|\mathbf{I}^P \wedge \mathbf{w}_j|}{\beta + |\mathbf{w}_j|} \quad (21)$$

where the ℓ -norm, $|\cdot|$, operator, is defined as

$$|\mathbf{x}| = \sum_{k=1}^N |x_k| \quad (22)$$

for an N -dimensional vector $\mathbf{x}$. The application of the ℓ -norm in Equation (21) allows the network to determine the proximity of the input vector to the vector range stored as the prototype in the F_2 node.

Applying Equation (22) to $\mathbf{w}_j$ of Equation (21) yields

$$|\mathbf{w}_j| = \sum_{k=1}^{2M} |w_{j,k}|. \quad (23)$$

Thus the activations for the uncommitted and committed nodes are

$$T_j(\mathbf{I}^P) = \begin{cases} \alpha_j \cdot |\mathbf{I}^P| & \text{for an uncommitted node} \\ \frac{|\mathbf{I}^P \wedge \mathbf{w}_j|}{\beta + |\mathbf{w}_j|} & \text{for a committed node} \end{cases} \quad (24)$$

As each new node is committed, a new uncommitted node is added to the system, with the α_j parameter set in accordance with Equations (17) and (18).

During operation, the F_1 layer searches for the F_1 node J with the highest activation with respect to input vector $\mathbf{I}^P$, using

$$T_j(I^P) = \max_j [T_j(I^P)]. \quad (25)$$

When the node with the highest activation is found, the orienting subsystem is activated to perform hypothesis testing on the chosen node, J .

The operations performed in the calculations of $T_j(I^P)$ in Equation (21) can be seen in the following figures. In Figure 4-12, a new vector, A , is compared to the existing prototype.

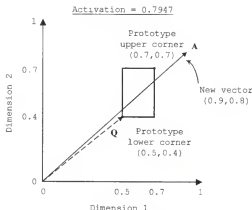


Figure 4-12. Activation response to new vector A .

Because the magnitude and direction of A exceed the extent of the rectangular prototype, the fuzzy-max of the vector with the upper corner of the prototype is (0.9, 0.8). The fuzzy-max is complement-coded in the prototype and input vector to allow processing with the fuzzy-min. The fuzzy-min of A with the lower corner of the prototype is the vector Q , at (0.5, 0.4). The activation of the prototype, from Equation (21), is

$$\begin{aligned}
 T_j(I^P) &= \frac{|I^P \wedge w_j|}{\beta + |w_j|} \\
 &= \frac{|0.5 \ 0.4 \ (1-0.9) \ (1-0.8)|}{\beta + |0.5 \ 0.4 \ (1-0.7) \ (1-0.7)|} \\
 &= \frac{0.5+0.4+0.1+0.2}{\beta+0.5+0.4+0.3+0.3} \\
 &= \frac{1.2}{\beta+1.5} \\
 &= 0.7947 \\
 \beta &= 0.01
 \end{aligned}$$

The activation for vector **A** is 0.7947.

Another vector is shown in Figure 4-13. This vector is closer to the neighborhood of the prototype. Taking the fuzzy-max of **A** versus the prototype upper corner yielded a new vector, **R**. **R** was complement-coded and used the same as the fuzzy-max in the previous figure. The resulting activation was 0.9603, which is higher than that of Figure 4-12, because the vector was closer to the prototype.

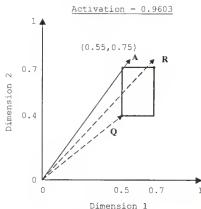


Figure 4-13. Activation response to closer vector **A**.

When the vector magnitude and direction cause it to fall below and to the right of the prototype, as in Figure 4-14, a new vector, **R**,

is created for the fuzzy-min, but no Q vector is created. This is because the components of vector A are below the levels of the corresponding components of the lower corner of the prototype. The fuzzy-min of A with the lower corner of the prototype is A . The activation for this vector is 0.5960. The activation is smaller than for the other vectors in the previous figures because it is further away from the prototype.

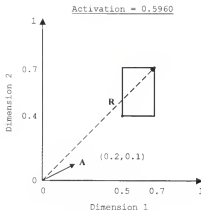


Figure 4-14. Activation response to small vector A .

The maximum activation for a vector occurs at any point within the prototype. The fuzzy-min decreases to the lower corner and the fuzzy-max increases to the upper corner. With the value for β set at 0.01, the activation here was 0.9934.

Orienting Subsystem

A network diagram of Fuzzy ART is shown in Figure 4-16. In addition to the layers shown in the attentional subsystem, there is a

set of connections shown for the orienting subsystem. The orienting subsystem performs a vigilance check.

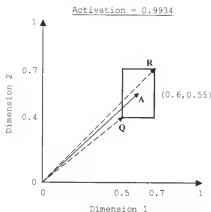


Figure 4-15. Activation response within prototype.

The template stored in the weights of the most active node is tested against the preprocessed input vector, $\mathbf{I}$. The vigilance test is

$$\frac{|\mathbf{I}^p \wedge \mathbf{w}_j|}{|\mathbf{I}^p|} \geq \rho^p \quad (26)$$

where ρ^p is the vigilance associated with the current input vector in the neural net training set. If the left-hand side of Equation (26) is greater than the right-hand side, then the vigilance test is satisfied. The F_2 node being tested is chosen as the category that best describes the input vector and is modified as required to learn any new information in the input vector, not previously part of the template.

If the vigilance test fails, then the activation of the node is reset to -1 and the vigilance of the next most active node is tested. This process continues until a committed node is found that satisfies

the vigilance criterion, or until the activations of the remaining nodes are below that of an uncommitted node, in which case a new node is committed to represent the input vector. When the vigilance test passes for a committed node, that node is said to resonate with the input vector. If an input vector resonates with the first tested node, the input vector is said to have direct access to that node.

Dynamic Operation

The full understanding of Fuzzy ART comes from both an architectural and a dynamic view of the network. As in other neural network architectures, the operation during training is different from the operation during testing or use. The major operations of the Fuzzy ART network are very similar in both training and testing, with the exception that the neurons are not updated or committed during testing. The operation of determining activation and checking resonance is the same in both training and testing.

Training: Two types of training will be considered, the standard neural network training that uses a prearranged training set and on-line training that responds to changes occurring during operation. The use of a prearranged training set will be called fixed-set training here to distinguish it from on-line training. In fixed-set training, the network will be trained on a predetermined training set consisting of a set of input cases for the network to learn. This training set will not change during the course of training. Fixed-set training is appropriate when training data is known in advance and new data is not being obtained on a continuous basis. Fixed-set training would be used to train the network on general turbine characteristics obtained from a statistical sampling of turbines operating under known conditions.

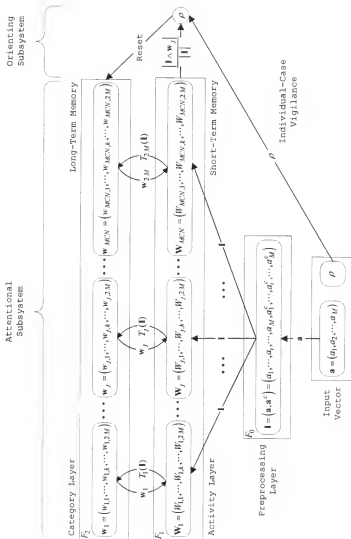


Figure 4-16. Fuzzy ART neural network interconnections

The network trained using fixed-set training could possibly serve as a general case system, suitable for an intended class of turbines, but would not be particularly sensitive to the detailed operating spectrum of a particular turbine installation.

The second type of training is on-line training. On-line training allows the neuronal weights to be updated as new information is presented to the network. In Fuzzy ART, changes are focused, isolated to one neuron at a time, within a range dependent on ρ . Individual category modifications can be made without affecting the rest of the long-term memory. This highly stable aspect of Fuzzy ART makes it attractive for fielded industrial applications where high dependability is expected over a long duration. Each change that occurs will not create a new neuron, but the memory should be sized based upon results of actual usage because new neurons will be created during training. As the engine condition changes, the network must continually update to learn the new vibration characteristics. Thus, a large number of new neurons could be generated. Ideally, the network could prune old neurons that no longer activate due to the changed spectrum. At each maintenance event, the system should be retrained to learn the spectrums resulting from the maintenance changes. This would allow the number of committed neurons to be lowered back to initial condition levels. The changes that occur to the neural net prototypes would be recorded for trending purposes, but could then be used to modify the training set, as required.

Once initially trained using fixed-set training, online learning does not require re-representation of the initial training set. If a novel vector arrives, from whatever point in a spectrum, the network can learn it immediately. Only one neuron will need to be committed

for the novel vector. If the new vector resonates with a prototype, it may cause a change in the prototype of a single neuron, but the remainder of the originally learned information does not change.

This ability to learn minor modifications in the spectrum without extensive retraining is a feature of Fuzzy ART that makes it suited for a neural trending system for a vibration analysis system, due to the resulting high-speed updates. As minor changes to the spectrum occur, the network can discern these changes, indicate that they have occurred, and learn the changes at a rate that should be suitable for real-time use.

Fixed-set training. The object of fixed-set training is to have the network completely settle down to a stable state. If, during the presentation of an entire epoch of data, no neurons change their weights, then the network is considered trained. The training-set is repetitively applied to the inputs to allow the system to create and modify its neuronal weights to learn the data. With a sufficiently small value of β , no new nodes will be dedicated after the first epoch. This fixes the maximum number of nodes at less than or equal to the number of cases in the input training patterns [34]. Unfortunately, there are many cases in the input training set, one for each frequency component. The network free variables, such as ρ , must be set to suitably compress the input information. The use of case vigilance also helps to reduce the number of committed nodes. The activity diagram in Figure 4-17 shows the operation during fixed-set training.

An input case from the training set is presented to the F_0 layer where it is complement coded and passed to the F_1 layer for activity detection. The F_1 layer checks the activity response of each node using Equation (24). The system then checks the vigilance of the most

active node using Equation (26). If the vigilance test passes, then the F_2 layer of that node is modified as required to learn any new information contained in the input case.

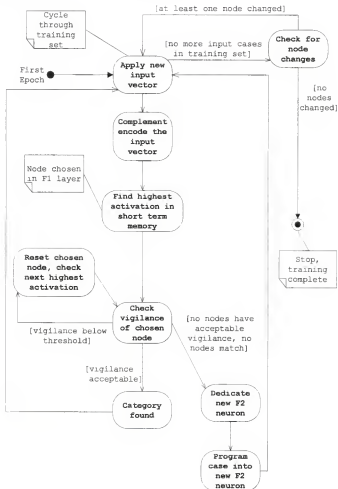


Figure 4-17. Activity diagram of Fuzzy ART learning.

If the vigilance test fails, the orienting system executes a reset function to force the network to check another node. The node with the next highest activation is then checked for resonance [32]. In implementation, the activation of the original node is set to -1, so that it will not be checked again and the remaining F_1 layer activations are checked for the next highest activation. The vigilance test is again performed on the newly chosen node. This process continues until either a suitable node is found or the activation falls so low that a previously uncommitted node is chosen. In Equation (26), the vigilance test is seen to be

$$\frac{|I^p \wedge w_j|}{|I^p|} \geq \rho^p.$$

The uncommitted node, w_j , has all its weights set to 1, thus the left-hand side of Equation (26) becomes

$$\begin{aligned} \frac{|I^p \wedge w_j|}{|I^p|} &= \frac{\sum_{j=1}^{2M} (I^p \wedge 1)}{\sum_{j=1}^{2M} I^p} \\ &= \frac{\sum_{j=1}^{2M} I^p}{\sum_{j=1}^{2M} I^p} \\ &= 1 \end{aligned} \tag{27}$$

Equation (27) shows that for an uncommitted node, the calculated vigilance is always 1. The vigilance test of Equation (26) thus becomes

$$1 \geq \rho^p$$

and always passes, because the pattern vigilance ρ^p is of the range [0,1]. If an uncommitted node is chosen due to highest activation, then

it will always be programmed using the values from the input case and become a committed node.

After the operations on the current input case are complete, the network training then continues with the next input case. The Fuzzy ART training stops when the presentation of an epoch of input cases results in no changes to the weights of the network. During training, the order of input cases can be changed from epoch to epoch to help ensure generalization within the network.

When a committed node passes the vigilance test, or an uncommitted node is chosen, the weights in the node are allowed to adapt to learn any new information contained in the input pattern. The weights of the chosen F_2 node are modified in the following manner:

$$\mathbf{w}_j^{(new)} = \gamma(\mathbf{I}^P \wedge \mathbf{w}_j^{(old)}) + (1-\gamma)\mathbf{w}_j^{(old)} \quad (28)$$

where

$$\begin{aligned} \gamma &= \text{system learning rate, } [0,1], \text{ set to 1 for fast learning} \\ \mathbf{w}_j^{(new)} &= F_2 \text{ weights after modification} \\ \mathbf{w}_j^{(old)} &= F_2 \text{ weights before modification} \end{aligned} \quad (29)$$

A notation is adopted to distinguish between new input information and old learned information. New information will be represented with the following notation: $\hat{\mathbf{w}}_j$, and old, learned, information with the following notation: $\tilde{\mathbf{w}}_j$. Thus, Equation (28) takes the following form:

$$\hat{\mathbf{w}}_j = \gamma(\mathbf{I}^P \wedge \tilde{\mathbf{w}}_j) + (1-\gamma)\tilde{\mathbf{w}}_j.$$

The system learning rate γ controls the speed at which the nodal weights adapt to changing inputs. There are three training options obtained by modifying the system learning rate. These options are slow learning, fast learning, and fast-commit slow-recode learning.

In slow learning, the system learning rate γ is set within the range $[0,1)$, always less than 1. Thus, the weights are not modified to totally encompass the new hyper-cube defined by $(I^P \wedge \bar{w}_j)$, but to move towards the new hyper-cube at a rate defined by γ . This type of learning is useful when trying to learn a pattern that is corrupted with noise. Noise is not learned immediately and patterns that persist for many applications will be gradually learned.

The fast learning option is most generally used. It requires setting $\gamma=1$ in Equation (28) resulting in the following learning rule:

$$\bar{w}_j = (\bar{I}^P \wedge \bar{w}_j) \quad \text{for fast-learning } (\gamma=1). \quad (30)$$

In fast learning, the node weight's hyper-cube is expanded until it encompasses the new pattern. Thus, when programming an uncommitted node, the node pattern will exactly match the input pattern, being a point in $2M$ -dimensional space. Reprogramming the same node will stretch the hyper-cube to fit modifications obtained through the fuzzy-min. This training method is more sensitive to noise because errors due to noise will be immediately learned.

The fast-commit slow-recode training option sets $\gamma=1$ while the node is uncommitted. After the node first becomes committed, γ is changed to a range $[0,1)$ to permit the learned information to change gradually in response to input patterns. This method has the advantage of fixing the initial hyper-cube of weights at first application, while being less sensitive to noise later in the programming.

On-line training. Placing the network into service to learn directly from the input cases as they occur allows the network to be more aligned with the actual data and avoids difficulties due to variation in sensor characteristics. The network is not given an epoch

of data to learn but is given either a single input case at a time or a set of input cases all derived from the subject at the same time. Appropriate averaging techniques must be applied to each data set to control the variance of the spectrum estimator. The system then creates classifications and commits neurons as required. The training executes exactly as in the fixed-set learning algorithm, but occurs continuously. It is important to plan for changes in the dynamic ranges of the data. If, over time, the input data range changes to cause clipping in the standardized [0,1] input range, the network performance will be degraded. Methods exist to dynamically alter the input ranges simultaneously with the Fuzzy ART learned weights. These techniques become important to increase the robustness of an installed implementation of the neural vibration monitoring system.

Testing and on-line operation. Testing the programmed neural net is accomplished by disabling the capability to encode new information and checking the category choice in response to test sets consisting of multiple input cases. During network operation, the input case will either resonate with a previously learned template, or will be sufficiently different from any learned templates that no resonance occurs and the input case is deemed novel. When a node is activated by the input case, but the vigilance test fails, a mismatch occurs. The amount that an input case is allowed to deviate from a learned template before mismatch occurs is controlled by the vigilance parameter ρ^P . The P superscript indicates that the vigilance is set on a case-by-case basis. In many applications, the vigilance is set as a constant, with each case having the same vigilance. In some cases, varying the vigilance can permit important information to be learned with more

detail than less important data. The less important data, such as noise data would be lumped together in larger classification regions.

During on-line operation, the network can either be allowed to continue to actively learn, or to use the neuronal weights that were set during training. An activity diagram showing Fuzzy ART testing and on-line operation is shown in Figure 4-18. The heavier lined items in the diagram shown the differences between the testing/on-line operation and the training operation.

Fuzzy ART is not a supervised neural net like Fuzzy ARTMAP, so the results of testing are not based on the resulting error of the network output with respect to a predetermined expected value. The primary figure of merit is the number of novel input cases that are discovered during testing. Ideally, all injected changes would be detected. The testing, to be described in later sections, included setting each spectral component to a level slightly higher than the change detection threshold used in the network design. If the change was detected, then that test passed. If the change was not detected, then the network exhibited a false negative response, and the test failed. The number of passing tests versus false negatives was used as the figure of merit. The final network design exhibited a ratio of 96% passes to 4% false negatives, when stimulated with changes with magnitude equaling the change detection threshold.

In application, larger number of novel cases may indicate that the network is overtrained or that the vigilance parameters may be set too high. The training and preprocessing parameters must be adjusted to minimize the number of novel cases that are generated in application, to minimize the amount of memory that must be provided for neuronal weights. The ability to be insensitive to small changes must also be balanced against the ability to detect important novel cases.

Testing with an actual turbine should provide data that may be used to adjust the training parameters to achieve an acceptable balance of detection capability versus memory requirement.

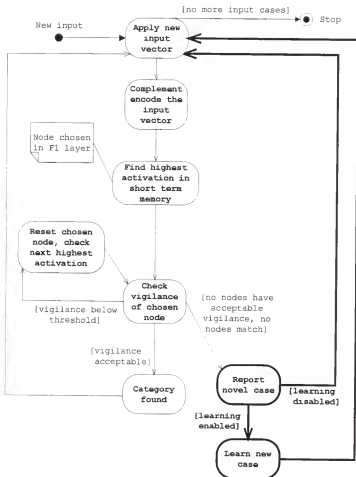


Figure 4-18. Activity diagram for Fuzzy ART testing and on-line operation.

The change detection threshold chosen for this research was 5% of full scale. This means that the network is designed to detect changes exceeding 5% of the maximum spectral amplitude range. The actual change detection threshold is arbitrary, as long as it surpasses the processing changes due to variance in the FFT estimator. With proper application of Welch's method of overlapping windowed FFTs, the estimator was accurate to within 3% of full scale. A change detection threshold of 5% of full scale was 67% larger than the 3% changes due to estimator variance. Thus, it should provide a reasonable minimum value for use in the detection of changes that result from physical turbine wear.

It is required that this 5% change be detected both for existing features that were already above the noise floor threshold and for features that previously were hidden in noise and change to exceed the noise floor threshold.

In order for the network to determine that a new input is novel, one of two situations must occur:

1. the activation $T_j(\vec{I})$ must be lower than that of any committed node, or
2. the vigilance test must fail for all nodes having activations higher than that of the first uncommitted node.

These situations assume that during testing there will be an uncommitted node, initialized in accordance with Equation (24), that will serve to indicate lack of activation potential. If no uncommitted node is provided during testing, then all short-term memory nodes would need to produce activations of $T_j(\vec{I})=0$ in order for situation 1 to fail. This is extremely unlikely, because it would require that there either be

1. no other nodes (other than the present node) programmed, or
2. two programmed nodes, with weights $w_1 = (0,1)$, $w_2 = (1,0)$, and the input $\bar{a}$ is either 0 or 1, a trivial case.

Having no uncommitted node available during testing causes vigilance checking for all nodes having activations greater than zero. Having an uncommitted node speeds up processing, by limiting the extent of vigilance testing to nodes having activation exceeding that of an uncommitted node. One uncommitted node was used.

Situation 1, the failure to have sufficient activation, is shown to generally require a large change in the input compared to the small amount (~5%) that is desired to be detected in the vibration monitor. From Equation (24), it is seen that the committed node activation is

$$T(\bar{\mathbf{i}})^{(committed)} = \frac{|\bar{\mathbf{i}} \wedge \bar{\mathbf{w}}|}{\beta + |\bar{\mathbf{w}}|} \quad (31)$$

and for an uncommitted node, the activation is

$$T(\bar{\mathbf{i}})^{(uncommitted)} = \alpha \cdot |\bar{\mathbf{i}}|. \quad (32)$$

The α term in Equation (32) is described in the section detailing the F_1 and F_2 layers, particularly Equations (17) and (18).

The norm of $\bar{\mathbf{i}}$ is

$$|\bar{\mathbf{i}}| = M \quad (33)$$

which stems from Equation (11) where $\bar{\mathbf{i}}$ can be seen to be

$$\bar{\mathbf{i}} = (\bar{a}_1, \bar{a}_2, \dots, \bar{a}_M, 1 - \bar{a}_1, 1 - \bar{a}_2, \dots, 1 - \bar{a}_M).$$

Summing the first and $(M+1)$ th terms yields

$$\bar{a}_1 + (1 - \bar{a}_1) = 1$$

and likewise for all the other of the M terms in $\bar{\mathbf{i}}$. Thus, the norm of $\bar{\mathbf{i}}$ can be seen to be

$$\begin{aligned} |\mathbf{i}| &= \sum_{i=1}^M [\bar{a}_i + (1 - \bar{a}_i)] \\ &= M \end{aligned} \quad (34)$$

Thus, the uncommitted node activation of Equation (32) becomes

$$T(\bar{\mathbf{i}})^{(uncommitted)} = \alpha M. \quad (35)$$

In order to choose an uncommitted node instead of a committed node would require that the activation of the committed node be less than the activation of the uncommitted node. This is shown as:

$$\begin{aligned} T(\mathbf{i})^{(committed)} &< T(\bar{\mathbf{i}})^{(uncommitted)} \\ \frac{|\mathbf{i} \wedge \bar{\mathbf{w}}|}{\beta + |\bar{\mathbf{w}}|} &< \alpha M \end{aligned} \quad (36)$$

The values of the components of α can be altered to achieve any value within the range (0,1), but general values used in practice are

$$\begin{aligned} \beta &= 0.01 \\ M_u &= 2M \end{aligned}$$

Substituting these values into Equation (36) using Equation (18) for α shows the test for choice of an uncommitted node instead of a committed node to be

$$\frac{|\mathbf{i} \wedge \bar{\mathbf{w}}|}{\beta + |\bar{\mathbf{w}}|} < \frac{M}{\beta + 2M} \quad (37)$$

meaning that if the left-hand side of Equation (37) is less than the right-hand side, choose an uncommitted node. Solving Equation (37) for $|\mathbf{i} \wedge \bar{\mathbf{w}}|$ yields

$$|\mathbf{i} \wedge \bar{\mathbf{w}}| < \left(\frac{\beta}{\frac{\beta}{M} + 2} + \frac{|\bar{\mathbf{w}}|}{\frac{\beta}{M} + 2} \right). \quad (38)$$

As β approaches 0, it is seen that the committed node activation must decrease below

$$|\bar{\mathbf{i}} \wedge \bar{\mathbf{w}}| < \left(\frac{|\bar{\mathbf{w}}|}{2} \right)$$

to cause selection of an uncommitted node instead of the committed node. This requires a 50 percent difference between the input $\bar{\mathbf{i}}$ and the previously stored information $\bar{\mathbf{w}}$, which is much greater than the 5% difference looked for in the vibration monitoring application. Increases in β cause the required difference to diminish. The activation-test algorithm can allow another committed node to activate if its stored information is within 50% the same as the input case. In this situation, the vigilance test must be relied upon to provide the necessary discernment to resolve input cases.

In Situation 2, the input has activated a neuron, but it must fail to satisfy the vigilance test in order to be flagged as a novel case. The changed feature must not resonate with any previously learned features. To fail the vigilance test requires that the vigilance calculated in the F_i node, $\rho^{(calc)}$, be below the case vigilance supplied with the input case, ρ^P . This test is

$$\rho^{(calc)} < \rho^P. \quad (39)$$

Substituting the definition of ρ into the left-hand side of Equation (39) yields

$$\frac{|\bar{\mathbf{i}} \wedge \bar{\mathbf{w}}_j|}{|\bar{\mathbf{i}}|} < \rho^P. \quad (40)$$

The fuzzy operator introduces sharp nonlinearities in the response of this ratio for different values of $\bar{\mathbf{i}}$ and $\bar{\mathbf{w}}_j$. This inequality will be investigated for the simple case of one value in the input case, $\hat{\mathbf{i}}$, and only one case learned by the template, $\bar{\mathbf{w}}$, of one neuron. This analysis will be expanded as the theory of resonance fields is introduced and

developed. Expanding Equation (40) for the analysis of one input case and one neuron yields

$$\frac{\bar{a}_1 \wedge \bar{w}_1 + \bar{a}_1^c \wedge \bar{w}_2}{\bar{a}_1 + \bar{a}_1^c} < \rho^p \quad (41)$$

Using Equation (34) allows Equation (41) to be written as

$$(\bar{a}_1 \wedge \bar{w}_1 + \bar{a}_1^c \wedge \bar{w}_2) < \rho^p \quad (42)$$

because $\bar{a}_1 + \bar{a}_1^c = 1$ in the denominator.

The node under investigation has been programmed with the original value of the feature, $\bar{a}$, and is being compared to the same feature, with a change in value, denoted by $\bar{a}$, where $\bar{a}_1 = \bar{a}_1 + \epsilon$. The template programmed into the node holds the original value of the feature:

$$\begin{aligned} \bar{w}_1 &= \bar{a}_1 \\ \bar{w}_2 &= \bar{a}_1^c \end{aligned}$$

For a negative-going change, where $\epsilon < 0$, the new input satisfies the inequality

$$\bar{a}_1 < \bar{a}_1$$

using the definition of the fuzzy-min. Equation (42) becomes

$$(\bar{a}_1 + \bar{a}_1^c) < \rho^p. \quad (43)$$

For a positive-going change, $\epsilon > 0$, the new input satisfies

$$\bar{a}_1 > \bar{a}_1.$$

Equation (42) becomes

$$(\bar{a}_1 + \bar{a}_1^c) < \rho^p. \quad (44)$$

Thus, a change in an input value will show up as a change in either the un-complement-coded or complement-coded value, but not both.

The amount of vigilance required to detect a certain amount of change will be determined for the 1-D case and generalized for

multidimensional inputs. The change in the input value will be represented as Δ , where $\Delta \in (0,1]$. The change will be referenced as a percentage of the full scale of the range. Thus, a 5% change will require $\Delta = 0.05$. For a negative-going change, Equation (43) can be written

$$\left[(\hat{a}_1 - \Delta) + \hat{a}_1^c \right] < \rho^P. \quad (45)$$

The inequality of Equation (45) is simplified as follows:

$$\begin{aligned} \left[(\hat{a}_1 - \Delta) + \hat{a}_1^c \right] &< \rho^P \\ \left[\hat{a}_1 - \Delta + (1 - \hat{a}_1) \right] &< \rho^P \\ 1 - \Delta &< \rho^P \end{aligned} \quad (46)$$

Therefore, for a 5% change the vigilance must be set as $\rho^P > 0.95$ in order for the change not to resonate and be deemed novel.

For the case of multidimensional inputs, Equation (40) will be expanded and simplified. Equation (40) expands to show all the internal information for the resonance check of a multidimensional input case versus an F_2 node $\hat{w}_J$, as follows:

$$\frac{|\hat{\mathbf{i}} \wedge \hat{\mathbf{w}}_J|}{|\hat{\mathbf{i}}|} < \rho^P$$

$$\frac{\hat{a}_1 \wedge \hat{w}_{J,1} + \dots + \hat{a}_I \wedge \hat{w}_{J,I} + \dots + \hat{a}_M \wedge \hat{w}_{J,M} + \hat{a}_1^c \wedge \hat{w}_{J,I(M+1)} + \dots + \hat{a}_I^c \wedge \hat{w}_{J,I(M+1)} + \dots + \hat{a}_M^c \wedge \hat{w}_{J,2M}}{\hat{a}_1 + \dots + \hat{a}_I + \dots + \hat{a}_M + \hat{a}_1^c + \dots + \hat{a}_I^c + \dots + \hat{a}_M^c} < \rho^P$$

The change that is being analyzed takes place in one dimension, $\hat{a}_I$, of the input case $\hat{\mathbf{a}}$. This test-change will be a positive-going change, thus affecting the complement-coded input value, $\hat{a}_I^c$. Continuing with the simplification yields

$$\begin{aligned}
& \frac{\sum_{i=1}^M \left(\tilde{a}_i \wedge \tilde{w}_{j,j} + \tilde{a}_i^c \wedge \tilde{w}_{j,j,M+1} \right) + \tilde{a}_I \wedge \tilde{w}_{j,j} + \tilde{a}_I^c \wedge \tilde{w}_{j,j,M+1}}{M} < \rho^p \\
& \frac{\sum_{i=1}^M \left(\tilde{a}_i + \tilde{a}_i^c \right) + \tilde{a}_I + \tilde{a}_I^c}{M} < \rho^p \\
& \frac{(M-1) + \tilde{a}_I + (1 - \tilde{a}_I)}{M} < \rho^p \\
& \frac{M-1 + \tilde{a}_I + 1 - (\tilde{a}_I + \Delta)}{M} < \rho^p \\
& \frac{M-1 + \tilde{a}_I + 1 - \tilde{a}_I - \Delta}{M} < \rho^p \\
& \frac{M - \Delta}{M} < \rho^p
\end{aligned} \tag{47}$$

As the dimensionality of the input case increases, higher vigilance is required to discern the same amount of change in any one dimension. For the 1-D case, Equation (47) is seen to equal Equation (46). To detect a 5% change with a 2-dimensional input case requires a case vigilance of

$$\begin{aligned}
& \frac{M - \Delta}{M} < \rho^p \\
& \frac{2 - 0.05}{2} < \rho^p \\
& 0.975 < \rho^p
\end{aligned} \tag{48}$$

For a 3-D case the case vigilance must exceed 0.9833 in order to cause a 5% change to be novel. For a 4-D case, the vigilance must exceed 0.9875 to recognize a 5% change. This amount of precision seems high, but it must be achieved in order to separate the large quantities of closely spaced information in the spectrum. With the use of 32-bit floating-point numbers, this amount of precision is achievable. Both the training and testing of the neural net must use the same values of ρ^p to allow the change to be detected.

Use of Adaptive Resonance Theory in Vibration Analysis

The information being analyzed in the Fuzzy-ART neural net is 4-dimensional and includes case vigilance. The information is organized as follows.

1. First dimension: frequency in spectrum
2. Second dimension: amplitude, derived from spectrum
3. Third and fourth dimensions: a priori feature separation, to be described in the subsequent sections.
4. Case vigilance: a priori vigilance levels, set on a vector-by-vector basis

The frequency and amplitude information is derived from the Fast Fourier Transform (FFT) of the turbine vibration spectrum. The a priori feature information is inserted dynamically, based upon the turbine rotational velocities, and the known features in the spectrum. The case vigilance is inserted dynamically for each case, based upon the amplitude.

In training, testing, and use of the network, signals are all preprocessed identically. The input vibration signal is sampled for a set period, converted to the frequency domain, and standardized to serve as neural network input. The a priori information is added on a spectrum-by-spectrum basis. The resulting matrix of input to the neural network has the following form shown in Equation (49).

The $\mathbf{A}$ matrix and $\mathbf{r}$ vector of Equation (49) can be used to hold the information from a single spectrum, or can hold data from multiple spectrums. The single spectrum case would be used for on-line processing of information, where the data would be continuously sampled and analyzed. A set of spectrums would be used to create $\mathbf{A}$ and $\mathbf{r}$ for training the neural net. When a set of spectrums is used, the spectral

information can be interspersed in different ways, the simplest being to append spectrum after spectrum. A random interleaving of individual cases from the spectrum data could also be used. To ensure repeatable results, and to allow isolated and controlled changes to be inserted for development and testing, a single estimate of the spectrum was used for initial analysis of the input preprocessing. When multiple spectrums were considered, the main concern was that the data be preprocessed to provide input that was biased identically from spectrum to spectrum.

$$\begin{aligned}
 \mathbf{A}, \mathbf{r} &= \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \mathbf{a}_3 & \mathbf{a}_4 \end{bmatrix}, \mathbf{r} \\
 &= \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ \vdots & \vdots & \vdots & \vdots \\ a_{p,1} & a_{p,2} & a_{p,3} & a_{p,4} \\ \vdots & \vdots & \vdots & \vdots \\ a_{P,1} & a_{P,2} & a_{P,3} & a_{P,4} \end{bmatrix}, \mathbf{r} \\
 &= \begin{bmatrix} \mathbf{F} & \mathbf{V} & \mathbf{J} \end{bmatrix}, \mathbf{r} \\
 &= \begin{bmatrix} F(1) & V(1) & J(1,1) & J(1,2) \\ \vdots & \vdots & \vdots & \vdots \\ F(p) & V(p) & J(p,1) & J(p,2) \\ \vdots & \vdots & \vdots & \vdots \\ F(P) & V(P) & J(P,1) & J(P,2) \end{bmatrix} \begin{bmatrix} \rho(1) \\ \vdots \\ \rho(p) \\ \vdots \\ \rho(P) \end{bmatrix}
 \end{aligned} \tag{49}$$

where

- $\mathbf{A}$ = matrix of input cases
- $\mathbf{r}$ = vector of case vigilances
- $\mathbf{F}$ = vector of frequencies from spectrum
- $\mathbf{V}$ = vector of vibration velocities from spectrum
- $\mathbf{J}$ = Matrix of a priori feature separations
- ρ_p = individual case vigilance
- P = number of input cases in $\mathbf{A}$

A block diagram describing the input preprocessing is shown in Figure 4-19. The resulting outputs are the vector components of the $\mathbf{A}$ matrix and the $\mathbf{r}$ vector. The theory behind the generation of each of these

vector components is discussed in the following sections, along with an investigation into the application of the FFT as the spectral estimator in this system.

Consistency of the Estimator

It is necessary to obtain a good estimation of the turbine spectrum for comparison in the neural network. The FFT is not a consistent estimator, due to variance in estimation. It also has bias that interferes with obtaining accurate data. There are techniques to improve both the bias and variance performance of the FFT, including Welch's method as will be described in this section. According to Jenkins and Watts [35], the main objectives in spectral estimation are high stability and high fidelity. They equate higher stability to lower variance and higher fidelity to the proximity of the estimated frequency response to the actual frequency response. The stability and the fidelity of the estimator used in this analysis were investigated, yielding techniques methods to improve the consistency of the estimator.

Jenkins and Watts also prefer a more empirical approach to improving the fidelity and stability of the spectral estimate, concluding that theoretical optimality solutions are not satisfactory. This is due to the stochastic nature of the spectral noise, the importance of unique spectral characteristics not being represented in the optimality equations, and arbitrary optimality solutions that may not satisfy different requirements. An empirical analysis of stability improvement was performed, using an application of Welch's method of overlapping windowed estimates of the frequency spectrum [36].

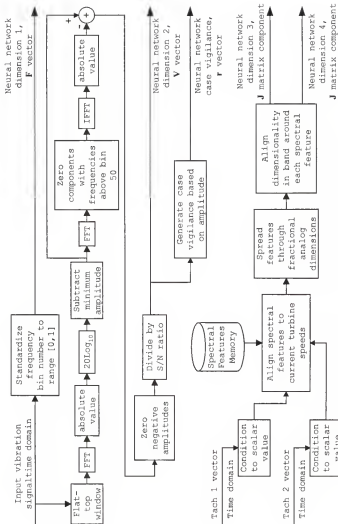


Figure 4-19. Neural network input signal preprocessing.

Fidelity. The estimates of the amplitude response in the spectrum are affected by both the estimator variance and the bias generated by the windowing function. Excessive variance in the estimator can affect the stochastically-determined mean of the resulting estimate, as will be shown using a box-whisker plot of the variance analysis. Estimate bias is also induced by the window applied to a time-limited section of data for analysis. The window function has additive sidelobe energy that affects the frequency response around each point in the response. As the length of the window of sampled data increases, the sidelobe energy of the window becomes more concentrated, approaching a periodic impulse train [37]. Thus, the FFT is asymptotically unbiased, with bias approaching zero as the window length increases.

It is important that the bandwidth of the window does not overlap features in the spectrum and bias their estimate. The flat-top window used in this research (described in the section on the **F** vector) has very low amplitude sidelobes (-70.42 dB), but has a relatively wide bandwidth of 8 FFT bins, or 4.89 Hz in this application. The minimum expected feature separation in this application is 50 Hz, representing the minimum operating speed of the gas generator, and hence its harmonics and sidebands induced in the other turbine-related spectral features. The bandwidth of the window is an order of magnitude less than the expected feature separation. Thus, the window bandwidth will not cause bias problems in the estimator.

In the experiments conducted to analyze the estimator stability, the interquartile range (IQR) of the estimations is determined. The IQR is defined in the box-plot discussion below. Each experiment placed the expected value of the estimation at a constant value of zero, by subtracting the expected value from the estimated value, thus

yielding a zero mean. Smaller IQRs, centered on the mean of the experiment group, indicate higher fidelity estimates of the expected value, along with indicating greater estimator stability.

Stability. A method to improve the stability of the FFT estimator is to average the frequency responses from uncorrelated data sets. In general, when K windowed sequences are averaged, the variance will be reduced by approximately $1/\sqrt{K}$ for the FFT, or $1/K$ for the periodogram [38]. To obtain the lowest correlation between data sets, the data sets cannot be overlapped. This presents two problems. The first is that an extensive amount of data must be obtained, increasing the possibility of sampling nonstationary data. The second problem is that if non-rectangular windows are used, transient signals may be missed in the areas between the main time-domain lobes of the windows. These problems can be minimized by using Welch's method, where nonrectangular windows are used and the sampling of the datasets is overlapped, as shown in Figure 4-20.

An empirical analysis was performed to determine optimum configurations of overlap and number of averaged sequences in the turbine signal application. A simulated turbine signal was used to control the level of the expected results for comparison with the estimated results and to obtain a suitably large signal, avoiding hardware constraints in the sampler. A set of signals was created, with each containing a sinusoid of known amplitude and a constant level of additive white Gaussian noise. The amplitude of the sinusoid was varied to obtain different signal-to-noise ratios within the amplitude range found in the actual turbine signal. The frequency of the sinusoid was varied through the frequency range of the turbine spectrum. These signals were processed through the same preprocessing

algorithm (described in the section on the $\mathbf{F}$ vector) as the actual turbine signals. This was done to include any variance-inducing properties of that processing in the analysis and to allow direct analysis of the results in the intended application.

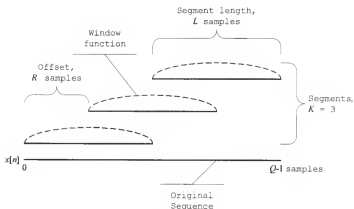


Figure 4-20. Partitioning of segments for Welch method processing.

The simulated input signal, $x[n]$, was constrained to be a maximum of 131072 samples long to help ensure stationarity in the sampling of an actual turbine. This amount of samples corresponds to 3.3 seconds of data sampling, during which time the turbine must remain at the same operating point. This maximum signal length was denoted as Q .

The input signal to the FFT had the following form:

$$x[n] = A_p \cos(\omega_p n) + \text{noise}(n), \quad 0 \leq n \leq (Q-1)$$

where

A_p = experiment amplitude

ω_p = experiment frequency

$\text{noise}(n)$ = additive white Gaussian noise,
zero mean, and constant variance

A flat-top window, $w(n)$, of length 65536 was applied to the data in each segment, yielding the resulting FFT equation

$$V(k) = \sum_{n=0}^{L-1} w(n)x[n]e^{-j2\pi kn/L}$$

A non-rectangular window, such as the flat-top window, decreases correlation in the overlap of the segments of the input sequence $x[n]$ because the extremes of the window are of lower amplitude than that of the main central lobe of the window. This allows the edges of the windowed signal to be overlapped, without overlapping the main window lobe, where more degradation due to correlation would occur due to the higher amplitude. This is an essential element of Welch's method. The flat-top window also reduces variance more than some other windows such as the rectangular, or Hanning window. The flat-top window has a variance reduction specification of 0.5015, with a 50% overlap, compared to 0.75 for a rectangular window and 0.5278 for a Hanning window.

The signal was processed through the neural network preprocessing and had a standardized spectrum similar to that shown in Figure 4-21 for a single amplitude and frequency combination. The sinusoid signal is visible above the noise floor.

Ignoring the effects of the narrowband window, which had been normalized to remove bias effects, the FFT of $x[n]$ has the following form, as seen in Figure 4-21

$$X[k] = f[\delta(k - \Omega) + \delta(k - N - \Omega)] + P(k)$$

where

$f[\cdot]$ = Standardization preprocessing

Ω = Digital frequency

L = Length of input sequence to FFT

P = Discrete noise process, simulating turbine spectrum noise floor

In the actual turbine signal, the features such as blade-pass vibration, sidebands and harmonics are generally distinct from one another. Thus, a single sinusoid with Gaussian noise provides an adequate simulation [39], especially considering that the actual signals encountered will vary slightly for each turbine, requiring a calibration effort at each installation.

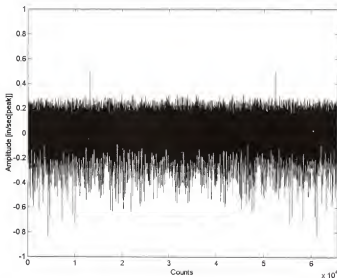


Figure 4-21. Example of simulated turbine signal.

The FFT was performed on segments of $x[n]$, having length L , where $L = 65536$ in this analysis. When multiple segments were analyzed, they were offset from each other by R samples. The number of FFT results that were averaged together was denoted by K . This is shown in Figure 4-20.

The maximum number of segments, K , that can fit within a sequence of length Q , with offset R , for a segment length of L , satisfies the inequality

$$(K-1)R + (L-1) \leq (Q-1). \quad (50)$$

With the sequence length set to $Q=131072$ and the segment length set to $L=65536$, the relation of Equation (50) becomes

$$R \leq \frac{65536}{K-1},$$

which governs the configurations for the Welch method testing.

Thirty-six different combinations of offset and numbers of segments were checked to find the best performing combination. For each combination, 120 different experiments were conducted, consisting of three sweeps through varying amplitude and frequency values. There were 45 groups of experiments, including the 36 different combinations, and multiple groups of single FFT groups. A single experiment consisted of creation of a signal, performing the windowing, FFT and neural preprocessing on all segments, and averaging all the resulting spectrums together. A new vector of random noise was created for each experiment. In each experiment, the difference between the estimate and the expected value was obtained.

The frequencies used in the experiments were approximately: 0.375, 0.5, 0.625, 0.75, and 0.875, where the value is from the standardized input range of $[0,1]$. At each of these frequencies, a pair of precise frequencies was generated, one directly centered on an FFT bin, and one in-between FFT bins, to induce the greatest possible degradation due to scalloping loss [38]. Thus, ten different frequencies were analyzed. The amplitudes of the sine wave were: 0.2, 0.4, 0.6, and 0.8 of a standardized $[0,1]$ range. The amplitude levels

were cycled through at each value of frequency. The entire set of amplitude/frequency experiments was repeated three times, yielding results for 120 experiments.

The Welch method configurations for the different experiment groups are listed in Figure 4-1. The group indexing of Figure 4-1 corresponds to the labeling in the analysis results of Figure 4-22 and Figure 4-23. Figure 4-22 shows the variances for each of the experiment groups of Figure 4-1. Figure 4-23 shows a box-whisker chart of the differences between the estimated value and the expected value.

The variance, σ_X^2 , in each group's estimates was calculated using

$$\sigma_X^2 = E[(X - \bar{X})^2] = \frac{1}{M} \sum_{j=1}^M (X_k - \bar{X}_k)^2$$

where

X_k = estimated amplitude of sinusoid

$\bar{X}_k$ = expected value of sinusoid, algorithmically set

$M = 120$, the number of estimates, or experiments, in each group

An improvement in the estimator stability was seen by examining the decrease in variance as the number FFTs being averaged increased. Referring to Figure 4-22 and Figure 4-1, the variance decreases rapidly as the number of FFTs increases in the following groups

Groups	Offset, R	FFTs averaged, K
27 to 31	16384	1 to 5
32 to 34	24576	1 to 3
35 to 37	32768	1 to 3
38 to 39	40960	1 to 2
40 to 41	49152	1 to 2
42 to 43	57344	1 to 2
44 to 45	65536	1 to 2

Table 4-1. Variance testing configurations.

Experiment group	Number of FFTs in ensemble, K	Sample offset, L
1	1	0
2	1	100
3	2	100
4	3	100
5	4	100
6	5	100
7	6	100
8	7	100
9	8	100
10	9	100
11	10	100
12	11	100
13	12	100
14	13	100
15	14	100
16	15	100
17	16	100
18	1	8192
19	2	8192
20	3	8192
21	4	8192
22	5	8192
23	6	8192
24	7	8192
25	8	8192
26	9	8192
27	1	16384
28	2	16384
29	3	16384
30	4	16384
31	5	16384
32	1	24576
33	2	24576
34	3	24576
35	1	32768
36	2	32768
37	3	32768
38	1	40960
39	2	40960
40	1	49152
41	2	49152
42	1	57344
43	2	57344
44	1	65536
45	2	65536

The lowest variance was experienced in group 34, corresponding to three FFTs averaged together, with a 24576-point offset. The required number of signals to sample for this combination of offset and number of FFTs is $L+(K-1)R=65536+2\cdot24576=114688$ samples, requiring 2.87 seconds to sample.

The notched box-whisker plot of Figure 4-23 provides a graphical comparison of the performance of each group of experiments. The box-whisker plot is a nonparametric, exploratory data analysis technique to simultaneously compare the distribution, central tendency, and variability of groups of experiments [40]. The box-whisker plot of a single experiment is shown in Figure 4-24. Figure 4-24 shows that the lower edge of the box is drawn at the lower quartile (25th percentile) and the upper edge of the box is drawn at the upper quartile (75th percentile). The height of the box thus encompasses the length between the upper and lower quartiles, also called the interquartile range (IQR). The term 25th percentile means that in the range of experiment results 25% of the results have a lower value than the result at the 25th percentile. The middle line of the box is drawn at the median [41]. The notch in the box shows a 95% confidence-interval estimate of the mean. Whisker lines extend from the bottom and top of the box and join the box to the farthest data point that is within 1.5 times the IQR from the edge of the box. Data points above and below the edges of the whiskers are termed outliers and are individually shown with the + symbol. A shorter box indicates more concentration around the median and provides a measure of estimator fidelity. The extent of the outliers above and below the box provides a measure of estimator stability, with a smaller extent, or range, indicating a higher stability.

The groups with an overlap of just 100 points, groups 2-17, provide poor performance due to the high correlation between samples. In these cases, the Welch-method overlapping induces a comb filter effect that is visible in the spectrum. The non-rectangular window provides no correlation reduction in these cases due to the short overlap.

As can be seen from the variance plot, the groups with the lowest variance are

Group	Offset, R	FFTs averaged, K
23	8192	6
25	8192	8
26	8192	9
30	16384	4
34	24576	3
45	65536	2

The best performance occurred in groups 30 and 34, with group 34 having the lowest variance. Group 30 appears to be the better choice because the outliers appear more evenly distributed. There is also less skew visible in the box-whisker plot; evident by the positioning of the median within the interquartile range represented by the box. From a computational load perspective, group 34 is preferable because only three FFTs are required to achieve essentially the same performance as the four FFTs of group 30.

The outliers of groups 30 and 34 all remained within ± 0.03 of the expected value. This corresponds to less than a 3% (of full-scale) change due to estimator variance. In the neural-network change-detection application, a 5% change is used in the decision to indicate that a vibration feature has changed. As evidenced by outlier range, two groups (1 and 2) that used a single FFT in the estimate produced

acceptable results, but unacceptable results were produced in seven other groups that used a single FFT (groups 18, 27, 33, 35, 38, 40, 42, and 45). Thus, the use of the Welch method improves the consistency of the estimator to yields consistently acceptable results, allowing the use of a 5% change as the decision point for the neural network. From this analysis, it is clear that the decision point for change detection should be made no smaller than 3%, based upon the performance of the estimator. Therefore, detecting a 5% change provides a measure of safety in the application. For an actual application, it would be necessary to analyze total system variance, including that of the sensor system and of the turbine in operation.

Summary of analysis of estimator consistency. Welch's method enables the FFT to become a more consistent estimator, improving both the fidelity and stability of the estimate. What appears to be the best performance for this type of signal was realized in group 30, with a 25% overlap length of 16384 points and an ensemble average of 2 FFTs. Essentially equivalent performance was provided in group 34 by a 37.5% overlap length of 24576 points and an ensemble average of three FFTs. The maximum variability of the estimator in these cases provides an estimate of the spectrum that was accurate to within 3% (of full scale) of the actual signal, even for outliers. This performance should be sufficient to allow a 5% change to be detected. The Welch method of variance reduction must be used on the data being processed for the neural net. Without the use of this variance reduction method, the worst case performance was 8.8% for the use of a single FFT.

Signal used for the neural network application development. In the analysis of the neural network algorithm, constraints in the amount of high-speed data that could be obtained in the digitizer required that an estimator using a single FFT be used for the analysis. The

variance problems were avoided during the analysis by using the same signal at each point in the analysis, but in application, the estimator must be stabilized through Welch's method of ensemble averaging. The use of a single, unchanging estimate of the spectrum provided a stable basis for development of the neural network application because it allowed comparisons of different neural-network configurations.

To develop the neural algorithm, the spectrum of the signal was directly modified to induce changes for detection during algorithm development. This would also have had to have been done for a spectrum obtained from an ensemble average set of spectrums, because the physical vibration amplitudes of the turbine could not be modified mechanically. Thus, the use of the estimate obtained from a single FFT permitted neural-network algorithm development, but must be modified in application to provide consistent estimation capabilities.

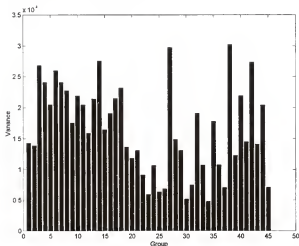


Figure 4-22. Variance of different configurations of FFT ensemble averaging.

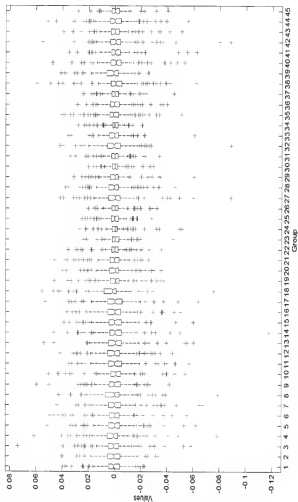


Figure 4-23. Notched box-whisker plot of experiment groups.

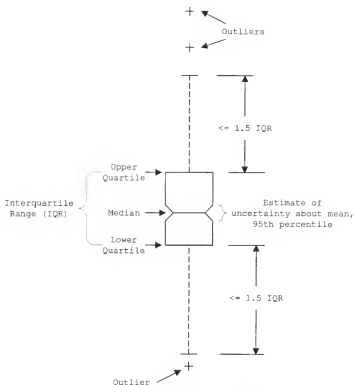


Figure 4-24. Description of box-whisker plot.

The $\mathbf{F}$ Vector

Analyzing from the perspective of a single spectrum being encoded into $\mathbf{A}$, the maximum extent in frequency of the spectral components determines the number of input cases, P . The $\mathbf{F}$ vector contains one value for each of the frequency components resulting from the FFT of the input vibration signal. The spacing of the frequency components, Δf , is

$$\Delta f = \frac{f_s}{N} \quad (51)$$

where

f_s = Sampling rate [samples/second]

N = Number of samples in the digital vibration signal.

In the research effort, the sampling rate, f_s , and number of samples, N , were chosen to take advantage of the available spectrum from the recorded tapes and were adjusted to use the rates obtainable by the hardware. Initially the data recorder tape outputs were viewed with a Spectral Dynamics spectrum analyzer to determine the general energy expanse in the signals. The sampling capabilities and available memory of the digitizing equipment were then investigated and used to determine the sampling rate.

The raw digital signal vector used to describe the processing to obtain neural network inputs is shown in Figure 4-25. The gas generator accelerometer signal was used for the analysis because the sensor is located closer to the high-pressure turbine sections therefore it may provide features that are more distinct for analysis. The vibration signal was sampled with a 16-bit analog-to-digital converter, yielding 65536 amplitude levels. A small portion of the raw digitized signal vector, showing signal detail is presented in Figure 4-26. The sampling rate was chosen as $f_s = 40026.06$ [samples/second], yielding an interval between time samples of $T = 1/f_s = 24.98372[\mu\text{sec}]$. The number of samples taken per digital signal vector is $N = 65536$ [samples]. These choices of f_s and N require a sampling duration of $N \cdot T = 1.637333$ [seconds] to obtain the entire digital signal.

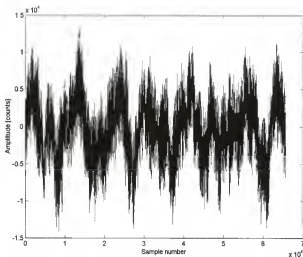


Figure 4-25. Raw digitized vibration signal, gas generator.

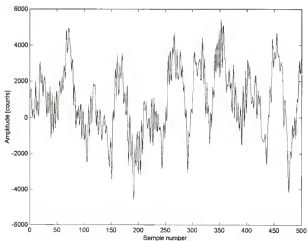


Figure 4-26. Raw digitized vibration signal, first 500 samples.

The amplitude data coming in from the digitizing equipment is contained in the range from -32767 to 32768 [counts] in magnitude. This digital representation was converted to [in/sec] for analysis. The data recorded on tape had the following calibration:

$$1381 [V[p-p]] = 4 [\text{in/sec[avg]}]. \quad (52)$$

The digitizer system had a peak input level of 3 [V[p]]; thus, it has a peak-to-peak range of

$$65536 [\text{counts}[p-p]] = 6 [V[p-p]]. \quad (53)$$

Converting the voltage level of the right-hand side of Equation (53) into units of [in/sec[p]] yields the following:

$$\begin{aligned} 65536 [\text{counts}[p-p]] &= 6 [V[p-p]] \\ &= 6 [V[p-p]] * (4 [\text{in/sec[avg]}] / 1381 [V[p-p]]) \\ &= 17.505 [\text{in/sec[avg]}] * [\pi [p-p/\text{avg}]] \\ &= 54.9936 [\text{in/sec}[p-p]] \end{aligned} \quad (54)$$

Thus, the desired conversion is

$$\begin{aligned} [\text{in/sec}[p-p]] / [\text{counts}[p-p]] &= 54.9936 [\text{in/sec}[p-p]] / 65536 [\text{counts}[p-p]] \\ &= 839.135 \cdot 10^{-6}. \end{aligned}$$

Using the sampling period and the unit conversions, the resulting input digital signal has the axes shown in Figure 4-27.

The next step in standardization of the input signal involves converting the signal to a frequency-domain representation. The calibrated input from the tape had the spectrum shown in Figure 4-28. Only the positive frequencies are shown for the spectrum plots because the absolute values of the negative frequency components are symmetric to absolute values of the positive frequencies. The input signal spectrum is shown in decibels in Figure 4-29.

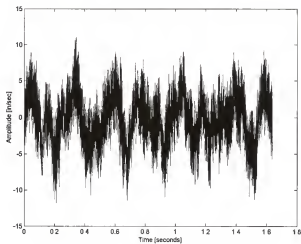


Figure 4-27. Calibrated input signal.

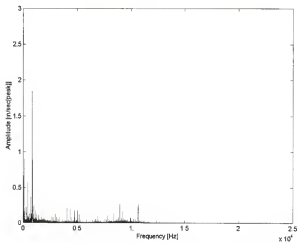


Figure 4-28. Calibrated input signal spectrum.

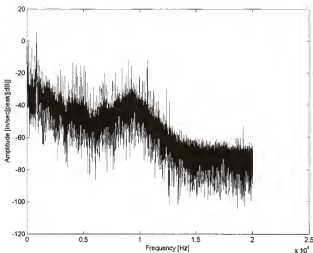


Figure 4-29. Input spectrum, dB.

Decibels are used to provide a logarithmic representation of signal level, with respect to a reference level. The decibel reference level in this case is 1 [in/sec[p]]. The decibel representation of the input velocity, v , is

$$v[\text{in/sec[p]}][\text{dB}] = 20 \cdot \log_{10}(v[\text{in/sec[p]}]) . \quad (55)$$

The term on the left of Equation (55) signifies that the measurement is in units of inches/second measured as unipolar peak information, in decibel representation. The decibel level is a logarithmic representation of signal level, with respect to a reference level. The reference level for the decimal representation in this research is 1 [in/sec[peak]]. The decibel representation of an input velocity, v , is

$$v[\text{in/sec[p]}][\text{dB}] = 20 \cdot \log_{10}(v[\text{in/sec[p]}])$$

where the notation for units of measure provides information that is important for correct interpretation of the vibration data.

The use of a logarithmic representation allows greater detail to be discerned in the spectral representation and facilitates an amplitude normalization technique that reduces dependence on transducer calibration. This normalization technique will be presented in the description for the contents of the input $\mathbf{V}$ vector.

The input spectrum showed support for the turbine vibration signals appearing to extend to approximately 15 kHz, with some features apparently occurring near 19 kHz. To satisfy sampling constraints in the digitizer and to capture the 19 kHz signal, a sampling rate of $f_s = 4002606$ [samples/second] was chosen, with a signal length of $N = 65536$ [samples]. Referring to Equation (51), this allowed a frequency spacing of $\Delta f = 0.6107492$ [Hz] between spectrum components. This provides a theoretical resolution of less than 1 [Hz] throughout the entire spectrum, up to the Nyquist rate of $f_{NY} = f_s/2 = 2001303$ [Hz]. Since the FFT generates a spectrum with symmetric properties (conjugate-symmetric real values and conjugate-antisymmetric imaginary values), only the positive frequencies were used to provide the amplitude information to the neural network.

The FFT generates the same number of output frequency components as the number of input samples, N , when N is a power of two and no zero-padding is used. Using only the positive frequencies from a single spectrum shows the number of input cases, P , in $\mathbf{A}$ of Equation (49) to be

$$P = N/2 = 32768 \text{ [input cases]}. \quad (56)$$

The $\mathbf{F}$ vector of Equation (49) is intended to hold the frequencies contained in the raw spectrum, linearly increasing from sample 0 to

sample P-1. The frequencies must be standardized to be used in the Fuzzy ART neural net. The inputs must all be of the range [0,1]. Therefore, the frequency inputs are all divided by the maximum frequency input. Although the representation of the spectrum in the range [0,1] compresses the frequency range together, very little information should be lost because 32-bit floating-point data units were used throughout. The ability to resolve closely spaced features in the spectrum is more difficult because of the short range into which the frequency range is compressed. To improve the resolution of closely spaced frequency components, the input space was augmented with more dimensions, controlled with a priori information, to separate the known features in the spectrum. The added dimensionality provides a metric in the new dimensions that preserves the separation between the components.

Since no more frequency information is available above the Nyquist rate, f_{NY} , the Nyquist rate is thus the maximum frequency input. The entries in the **F** vector are

$$F_p = \frac{p \cdot \Delta f}{f_{NY}}, \quad p = 0..(P-1) \quad (57)$$

The **V** Vector

The **V** vector holds the amplitude information for the input to the neural network. The amplitude information is taken from the spectrum of the input signal. The spectrum is created with the FFT.

The FFT by itself is not accurate enough to use for amplitude measurements; therefore, the input time-domain signal must be windowed to increase the accuracy. One of the best window functions for ensuring accurate amplitude is the P301 flat-top window, because it

provides a very accurate, low-bias, amplitude representation, essentially eliminating scalloping losses. It also has low sideband energy, helping to reduce bias in the estimate, and has good variance-reduction capabilities when used with overlapping FFTs.

The flat-top window was invented by Ron Potter at Hewlett-Packard. It is so named because the amplitude of the energy of resulting spectral components appears flat across a small frequency range around the center of the component. The mathematical function for generating the flat-top window is

$$w(n) = a_0 + 2 \sum_{k=1}^3 a_k \cos 2\pi k \left(\frac{n}{N} \right), \quad n = 0..(N-1) \quad (58)$$

where

$$a_0 = 0.9994484$$

$$a_1 = -0.955728$$

$$a_2 = 0.539289$$

$$a_3 = -0.0915810$$

$$N = 65536[\text{samples}], \text{ in this case.}$$

The ability of the flat-top window to create accurate amplitude information is illustrated in the following figures. A sine wave was created that is controlled to have a frequency that is exactly between the centers of two adjacent FFT frequency components. This waveform was created using the same sampling frequencies and the number of samples as in the vibration analysis signals. There are 32768 frequency components in the positive frequency side of the FFT output. A frequency was chosen that is between FFT component 16000 and FFT component 16001. This frequency is at imaginary frequency component 16000.5 and has a frequency 9772.2927 Hz. Figure 4-30 shows the generated sine wave. The amplitude of the sine wave shows some variation due to sampling at f_s [samples/sec].

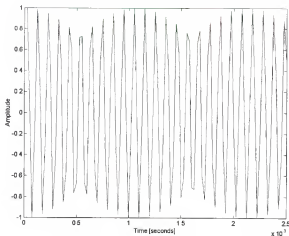


Figure 4-30. Raw sine wave, with frequency between FFT components.

The flat-top window that was generated to cover the time-domain sine wave signal appears in Figure 4-31. Applying the flat-top window to the time-domain sine wave signal yields the waveform of Figure 4-32. Taking the FFT of both the non-windowed and the windowed sine waves results in Figure 4-33 and Figure 4-34.

The amplitude of the sine wave energy in the spectrum should be 1, corresponding with the amplitude of the sine wave in the time-domain. In the non-windowed FFT of Figure 4-33, the sine wave amplitude is 0.6367, whereas the flat-top windowed FFT of Figure 4-34 has a nearly perfect amplitude level of 0.9994. The flat-top window error is less than 0.009 dB. The flat-top window outperforms most other window functions in amplitude accuracy between bins (scallop loss [38]) including the triangle window (3.92 dB loss), rectangle (3.07 dB loss), Hanning (0.86 dB loss), Hamming (1.78 dB), and Minimum 4-Sample

Blackman-Harris (0.83 dB loss). The flat-top window is applied to each input vibration signal before processing with the FFT.

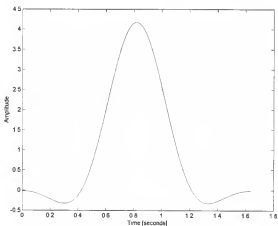


Figure 4-31. P301 Flat-top window.

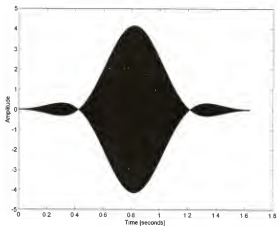


Figure 4-32. Windowed sine wave.

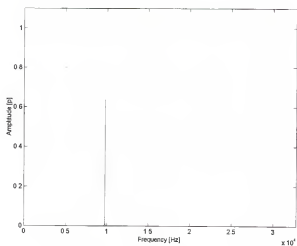


Figure 4-33. FFT of raw sine wave.

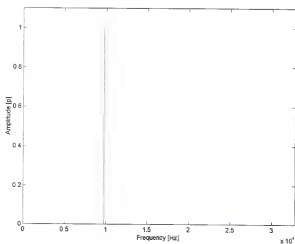


Figure 4-34. FFT of flat-top windowed sine wave.

The raw spectrum of the whole signal in Figure 4-29 shows wide variations in the baseline of the noise floor. This variation would result in the amplitude ranges of the noise floor overlapping with the signal amplitude ranges in certain types of neural network architectures. Some of the waviness in the spectrum is attributable to operations in the data recorder. While the absolute amplitude of vibration components is important, it is desired to use the neural network to detect changes in the component amplitude over time. This can be performed in many ways, but the use of neural network provides a way to compress the spectral information in a controlled manner, thus minimizing the amount of memory needed to represent the spectrum. A 5-dimensional representation of the spectrum in this research required 228 nodes of 32-bit floating-point numbers, requiring 9120 bytes of storage. The entire 32768-point spectrum, if stored as 16-bit numbers, would require 65536 bytes of memory. Thus, an 86% reduction in memory requirements is provided using the neural network. Also considering the capability to automatically learn new information, the neural net provides attractive capabilities to the trending application. If the actual amplitude of the components is needed, it can be easily determined by using the tachometer data and blade counts to index into the raw spectrum. To remove the bias due to the changing noise floor requires that the statistical means of the noise floor be extracted from the raw spectrum. With the noise baseline flattened, the peaks extending above the noise baseline should correspond to features of interest associated with mechanical components on the turbine.

The mean value of noise throughout the spectrum was obtained by FFT-filtering the input spectrum. The FFT of the input spectrum was taken and the higher frequency information was cut out, leaving the low frequency movement of the noise floor. The logarithmic representation

of the spectrum contains both positive and negative values, so the spectrum shifted to make it unipolar positive, by subtracting the value of the most negative element, in order to facilitate manipulation with absolute values.

Taking the FFT of the unipolar spectrum of Figure 4-29 yields Figure 4-35. Note that this is not the spectrum of the input time-domain signal, but it is the FFT of the FFT of the input time-domain signal. Because the noise baseline is slowly changing with respect to the higher frequency information contained in the spectrum, the low frequency noise baseline information was removed from the input spectrum. The following description shows the calculation of the baseline from the residual information left in the FFT of the FFT after zeroing the high frequency information. A faster method to accomplish the same thing would just zero the low frequency information in the FFT of the FFT, without incurring extra processing to perform subtractions of the mean throughout the spectrum.

Subtraction of the noise baseline in logarithmic units is equivalent to division by the noise baseline in linear units. Because the logarithmic units are used to provide more detail in the spectrum, performing a subtraction of logarithmic units is used instead of division in linear units as a possible computational cost reduction method. Division operations can take more time to perform than subtraction operations, especially if performed in software. It may slightly offset the computational cost of converting the spectrum values to logarithmic numbers.

Removing most of the upper information in the FFT shown in Figure 4-35 to leave the information from the more slowly changing noise-baseline intact yields the abbreviated FFT of the FFT of the input signal, shown in Figure 4-36.

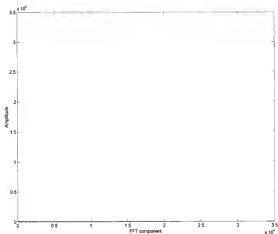


Figure 4-35. FFT of FFT of input signal. The DC level causes the upper frequency information to appear near zero amplitude.

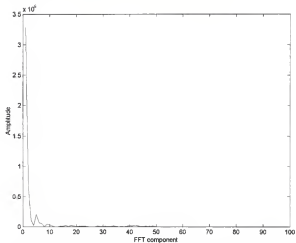


Figure 4-36. FFT of FFT of input signal with higher components zeroed. Note small extent of non-zero information compared to the complete 32768-component FFT.

Taking the inverse-FFT of the abbreviated double-FFT of the input spectrum yields the mean of the noise-baseline in the input spectrum, shown in Figure 4-37. The result is a low-pass filtered representation of the spectrum.

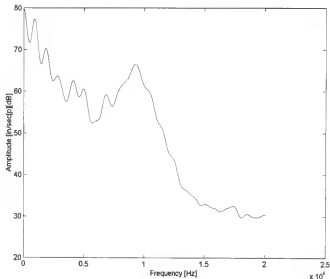


Figure 4-37. Input spectrum noise-baseline.

Subtracting the noise-baseline of Figure 4-37 from the original spectrum of Figure 4-29 yields the leveled spectrum of Figure 4-38. The spectrum in Figure 4-38 was also adjusted to remove all negative amplitude components. The neural trending application is primarily concerned with signals that are increasing over time, or at least having energy greater than the noise baseline. Information pertaining to short spectral excursions below the noise baseline is not required.

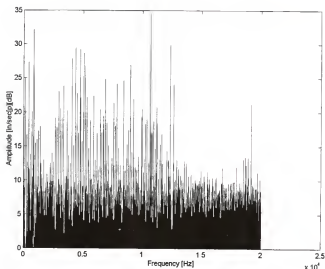


Figure 4-38. Input spectrum with noise-baseline standardized.

With the use of logarithmic quantities, the subtraction of the mean also performs a standardization function that removes linear transducer and calibration errors. Showing the calibration constant as C and the transducer error as ε , then the subtraction of the decibel mean from the decibel input spectrum has the following effect:

$$\begin{aligned}
 &20 \log_{10} \{C \cdot \text{SNR}(k) [b(k) + \varepsilon]\} - 20 \log_{10} \{C [b(k) + \varepsilon]\} \\
 &= 20 \{ \log_{10} C + \log_{10} \text{SNR}(k) + \log_{10} [b(k) + \varepsilon] - \log_{10} C - \log_{10} [b(k) + \varepsilon] \} \\
 &= 20 \log_{10} \text{SNR}(k)
 \end{aligned} \tag{59}$$

where

C = calibration constant

ε = transducer gain error

k = spectrum component

$\text{SNR}(k)$ = signal to noise ratio at spectral component k

$b(k)$ = noise-baseline at spectral component k .

Thus, subtracting out the mean of the decibel vibration level yields just the signal to noise ratio $SNR(k)$ at each spectral component. This signal to noise ratio is taken at each component in the spectrum, it is not an overall signal to noise ratio. In this case, the actual spectrum is characterized by the vector of $SNR(k)$ values.

The ability to eliminate gain error in calibration and in the transducers is a benefit stemming from removal of the noise-baseline. The actual decibel extents of the vibration features are not changed in this manner, but are more easily comparable to the other features in the spectrum using a neural network. The noise-baseline waveform is assumed constant from experiment to experiment although recalculated for each spectrum. In implementation, it may be enforced that a constant noise-baseline waveform be subtracted from the data, instead of recalculating the noise-baseline at each iteration.

To serve as input to the Fuzzy ART network, the amplitude should be standardized to fall within the range $[0,1]$. This requires that a maximum spectrum-component amplitude be determined and that all spectral components be scaled with respect to this maximum. The signal to noise ratio of the data recorder was specified by the manufacturer as 48 dB. It is seen from Figure 4-38 that the calculated amplitudes do fall within this range. A survey of the amplitudes from various taped data sets was performed and the maximum signal-to-noise ratio did not exceed approximately 34.9 dB. The maximum signal-to-noise ratio from the data recorder manufacturer was used as the maximum expected signal to noise ratio, SNR_{max} . All noise-baseline leveled spectral components were divided by SNR_{max} to yield the completely standardized input to the neural network, as shown in Figure 4-39. This figure also

shows the frequency dimension standardized to a $[0,1]$ range by dividing by the maximum frequency throughout the range.

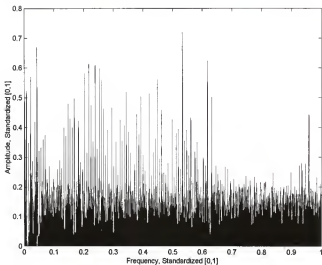


Figure 4-39. Standardized input signal.

The amplitudes of the standardized input signal of Figure 4-39 are used in the input vector V of Equation (49).

Case Vigilance Vector

The utility of the case vigilance vector is seen from training the neural network without using case vigilance. Without case vigilance, most of the resources of the network are devoted to learning the characteristics of the low level noise in the input spectrum. Because the spectral information can extend up to approximately 40 dB above the majority of the low-level noise, it was possible to reduce the amount of network resources devoted to learning the noise by

modulating the vigilance parameter. The effects of different values of vigilance on the neural representation of the spectrum are shown below, along with an example of the problem encountered when learning the entire spectrum, including the noise-floor information, with a high vigilance. An approximation of the noise floor threshold is made, and the vigilance then modulated for each component, depending on the component amplitude with respect to the threshold. Components with amplitude below the threshold are assigned a lower vigilance, while components with amplitudes above the threshold are assigned a higher vigilance. The performance of the network with this modification to the input was tested, with the results shown. Approximately $\frac{1}{4}$ of the nodes (74%) required to represent the spectrum were eliminated.

Using an initial input to the Fuzzy ART neural network of the form shown in Equation (60).

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} \\ \vdots & \vdots \\ a_{p,1} & a_{p,2} \\ \vdots & \vdots \\ a_{P,1} & a_{P,2} \end{bmatrix} = \begin{bmatrix} \mathbf{F} & \mathbf{V} \end{bmatrix} = \begin{bmatrix} F(1) & V(1) \\ \vdots & \vdots \\ F(p) & V(p) \\ \vdots & \vdots \\ F(P) & V(P) \end{bmatrix}. \quad (60)$$

This initial input also does not include the a priori classification vector. The utility of the feature separation vector, $\mathbf{J}$, will be described in the Feature Separation Vector section.

Without using case-by-case vigilance values, an overall vigilance value, $\bar{p}$, must be chosen for the whole training set. This vigilance will be termed set vigilance, to distinguish it from case vigilance.

Low values of vigilance (~0.2 to 0.5) will result in rough classification. Conversely, higher values of vigilance (~0.95) will result in a proliferation of categories as the neural net attempts to learn the noise with the same vigilance as the features. The effects

resulting from using different set vigilance values are shown in the figures that follow.

When two-dimensional inputs are used in a Fuzzy ART system, the resulting classification regions, or prototypes, become rectangles in a two-dimensional plane. As the long-term memory of the F_2 layer is modified to learn the input spectral features, the classification rectangles are committed and enlarged as needed to encompass the input features. From Equation (12), the weights in the F_2 layer are shown as

$$\mathbf{w}_j = (w_{j,1}, \dots, w_{j,k}, \dots, w_{j,2M}).$$

The first half of the weights, $(w_{j,1}, \dots, w_{j,k})$, represents information that is modified by taking the fuzzy-min of the existing weights with the new information from the uncomplemented section of the input vector, $\mathbf{I}$. The second half of the weights of $\mathbf{w}_j$ represents information that is modified by taking the fuzzy-min of the existing weights with the complemented section of the input vector. As an individual long-term memory node, $\mathbf{w}_j$, learns a series of patterns, the pattern stored in $\mathbf{w}_j$ becomes a compressed version of all the input patterns, $\mathbf{I}(x)$, that resonate and modify, or code, the node. These resonating and coding patterns, having index x , will be grouped in the set R , represented by

$$\mathbf{I}(x), \quad x \in R.$$

The set, S , of all indices, x , in the epoch of input patterns is

$$S = \{1, 2, \dots, p, \dots, P\}.$$

Thus the complete input vector of $\mathbf{I}$ values is

$$\mathbf{I}(x), \quad x \in S.$$

R is a subset of S , or

$$R \subseteq S.$$

The set, R , consists of all indices, x , in S for which the associated input pattern resonates and codes the individual node w_J , shown as

$$\{x | w_J \text{ resonates and codes } I(x)\}$$

The pattern stored in w_J becomes

$$w_J = \bigwedge_R I(x), \quad x \in R, \quad (61)$$

using the fast-learning rule of Equation (30). It is highly unlikely that all input cases in the input training set will resonate with the J node of Equation (61), thus in general R is a proper subset of S , or $R \subset S$. Each I vector has the form shown in Equation (11). Using Equation (11) in Equation (61) yields the representation shown in Equation (62)

$$\begin{aligned} w_J &= \bigwedge_R I(x), & x \in R \\ &= \left(\bigwedge_R a(x), \bigwedge_R a^c(x) \right), & x \in R \\ &= \left[\bigwedge_R a(x), \left(\bigvee_R a(x) \right)^c \right], & x \in R \\ &= (u_J, v_J^c) \end{aligned} \quad (62)$$

where

$u_J = \bigwedge_R a(x)$ the fuzzy-mins of all resonating and coding inputs $a(x)$

$v_J^c = \left(\bigvee_R a(x) \right)^c$ the complement of fuzzy-maxs of input $a(x)$ vectors.

For a two-dimensional input, as in Equation (60), the u_J vector contains the lower left vertex of the classification rectangle. The v_J vector contains the upper right vertex of the classification rectangle. This relationship is shown in Figure 4-40. The 2-D point represented

by the u_j vector changes toward point (0,0) as information is learned that has a lower numerical value in one or both of these dimensions than the stored information. Similarly, the point represented by the v_j vector grows toward point (1,1) as higher numerical value information is learned. For the processing of the Fuzzy ART system, the v_j vector is stored in the complemented form to represent increasing numerical value as decreases in the stored value. In the spectrum analysis application, Dimension 1 corresponds to the standardized frequency and Dimension 2 corresponds to the standardized amplitude.

Using a low set vigilance of $\bar{\rho}=0.2$ results in the classification regions of Figure 4-41, shown superimposed upon the standardized input spectrum of Figure 4-39.

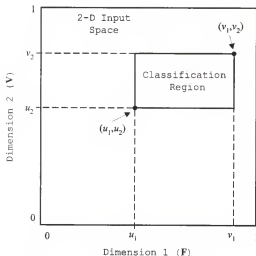


Figure 4-40. Classification Region Diagram.

In Figure 4-41 it is seen that only two classification regions were created, using a set vigilance of $\bar{\rho}=0.2$. The neural net required 111 epochs to perform the classification, taking 10 seconds on a dual-Pentium Pro, 200MHz, 256k-cache machine with 128 MB RAM running Windows NT. Two classification regions are inadequate to distinguish features; thus, a higher vigilance must be used. Figure 4-42 shows the results of setting the set vigilance to $\bar{\rho}=0.5$.

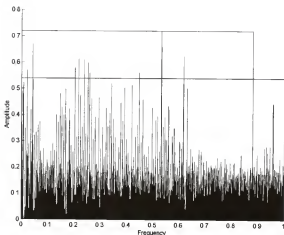


Figure 4-41. Neural classification regions, $\bar{\rho}=0.2$.

There are four classification regions in Figure 4-42, requiring 402 epochs and 61 seconds to train. The neural net is starting to distinguish noise from spectral peaks, as is seen by the rectangle in the lower right. However, this amount of set vigilance is too low to perform meaningful classifications on this data. It was provided to illustrate the effect of the vigilance parameter. In order to perform

the desired detection function, the vigilance is set based upon the amount of change that is to be detected.

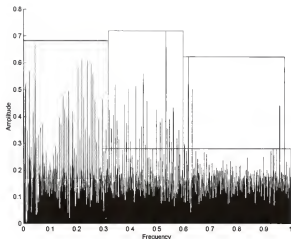


Figure 4-42. Neural classification regions, $\bar{\rho}=0.5$.

Raising the vigilance will cause more nodes to be committed. Using a single value of vigilance for all the information in the spectrum, including the information that appears to be primarily low-level noise, will cause an excessive number of nodes to be dedicated. Many of the nodes would be wasted mapping the noise information. A method was developed to set the vigilance of the noise to a low value, while using the higher vigilance on the higher amplitude signals. Thus, the higher vigilance, required by design to detect the changes, can be used while minimizing the number of nodes committed.

From Equation (48) it is seen that the vigilance required to detect a 5% change in one of the input dimensions is $\rho > 0.975$. Changing

the set vigilance to $\bar{\rho}=0.976$ results in the classification regions of Figure 4-43. For clarity, Figure 4-44 shows just the classification regions of Figure 4-43, without superimposing the spectrum information. These classification regions appear small, but the small size is required in order to resolve a 5% change in the 20kHz of frequency data represented in the range [0,1]. If the detection of larger changes is desired, the classification regions will grow accordingly and fewer nodes will be required.

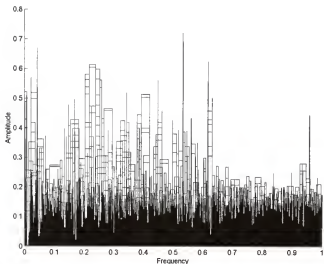


Figure 4-43. Neural classification regions, $\bar{\rho}=0.976$.

Using a higher vigilance value causes more nodes to be committed, which is excessive because most of the nodes were dedicated to learning noise information. A technique to reduce the number of nodes is described below. Using a set vigilance of $\bar{\rho}=0.976$ commits 891 nodes,

or classification regions, taking three epochs and 86 seconds. While the classification regions may be able to resolve a 5% change in a spectral component, the noise information is learned to the same degree as the spectral peaks. Having learned the noise to this degree means that every change in the noise exceeding 5% would be flagged and recorded as a change by the neural trending system. There is no reason to learn the noise information to the high degree of vigilance that is shown in Figure 4-44. Modifying the case vigilance with respect to the amplitude of each standardized spectral component allows Fuzzy ART to classify the noise using large classification regions, while learning the peaks closely.

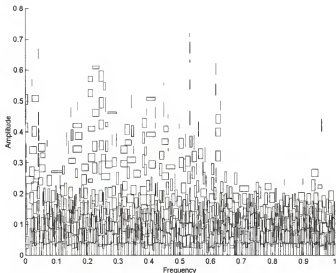


Figure 4-44. Neural classification regions, $\bar{\rho}=0.976$, no spectrum.

In order to use case-by-case vigilance, where each input vector is given a distinctive vigilance setting, an appropriate cutoff point for noise versus the desired spectral information is required. Looking at the standardized input spectrum of Figure 4-39 shows that an amplitude cutoff point of 0.1 would be too low because many noise components have higher peaks than 0.1. Similarly, an amplitude cutoff point of 0.4 is well above the apparent noise energy, but 0.4 would likely be too high because some of the lower spectral peak information is lost. A guess of a suitable value would be sufficient for demonstrating the change detection algorithm, but a statistical approach was developed as a first approximation to the most suitable value. Taking an approximation to the cumulative distribution function (CDF) of the spectrum amplitudes provided a somewhat repeatable estimate of what amplitude the noise floor stops and the higher amplitude signals began. The pseudo-CDF of the standardized spectrum is shown in Figure 4-45.

The CDF of the standardized spectrum shows a pronounced knee in the distribution. The energy due to noise accumulates almost linearly in the CDF until the bending point at the knee where the higher amplitude spectral peaks begin to accumulate. The CDF does not fit to a normal distribution (nor other distributions tried), primarily because of the amount of energy remaining above the knee, in the range (0.252,1).

A line was drawn tangent to the edge of the knee to find the midpoint of the knee, occurring at approximately (0.252, 0.935). At this point, the noise energy begins to decrease and the distribution contains higher-level amplitude components. The bulk of the low-level noise is contained below this point. This cutoff level is a first approximation that may be preferable to choosing a setting without

knowledge of the distribution. Lower settings will commit more nodes and higher settings commit fewer nodes. In practice, the setting will depend on important vibration threshold levels in the machinery vibration, on the existing noise characteristics and on the amount of memory available in the system.

Above an amplitude of 0.252, the spectral peaks tend to dominate the standardized spectrum. By this measure, it is seen that approximately 93.5% of the information in the standardized spectrum is deemed unimportant noise information.

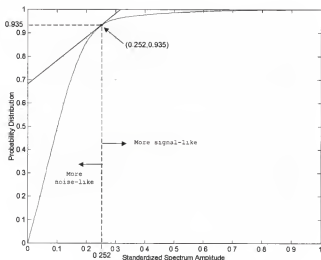


Figure 4-45. Pseudo-cumulative distribution function of standardized spectrum.

The Fuzzy ART noise-threshold amplitude (v_{min}) was set to the value corresponding to the edge of the knee information in this experiment, or $v_{min} = 0.252$. Below v_{min} , the case vigilance was set to

0.1 and above v_{min} , the case vigilance was set to 0.976, as indicated by Equation (48). Using these settings, it is desired to ignore most noise energy and to discern amplitude changes of greater than 5% in the spectral peaks. As seen in Figure 4-46, the noise floor information is classified with fewer nodes than the higher-level signals. Other levels of noise-floor threshold would commit more or less nodes, but this level is sufficient for the analysis of change detection capabilities. The Fuzzy ART neural network was trained with these values of v_{min} and case vigilance, with the results shown in Figure 4-46 and Figure 4-47.

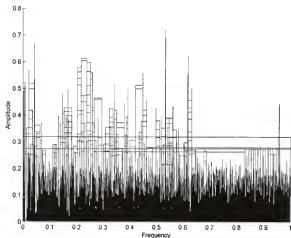


Figure 4-46. Classification of components using case vigilance.

It is seen in Figure 4-46 and Figure 4-47 that the higher amplitude signal information is learned to a greater vigilance than the noise information. This neural classification required 650 epochs to

train, taking 2750 seconds, and produced 152 classification regions, or nodes.

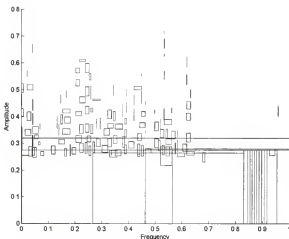


Figure 4-47. Classification regions using case vigilance.

Individual frequency component testing. The theory governing detection of changed components, presented in the section concerning the orienting subsystem, was tested to determine if a 5% change in amplitude is detectable. A 5% amplitude increase was added to the spectral energy associated with the 11th to 16th blade set. The 11th to 16th rotor stages of the LM2500 each have the same number of blades, thus forming a large spectral peak. The classification regions for the unchanged 11th to 16th blade set feature are shown in Figure 4-48 and Figure 4-49.

The classification regions of the network (a small portion of which is shown in Figure 4-48 and Figure 4-49) have adapted to encompass every point in the spectrum. All input vectors are contained

in the set of classification regions. One point, located at a standardized frequency of approximately 0.53346 appears to not have a classification region associated with it, but actually has a single point classification region that does not readily show up on a paper plot. It is unlikely that any single point classification regions will exist when trained with a training set larger than a single spectrum, as was done in this initial example. Figure 4-50 shows the spectral component that was increased in amplitude.

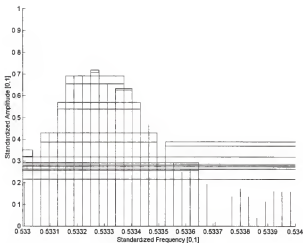


Figure 4-48. 11th to 16th Blade set energy before 5% change.

When the neural net that had been trained earlier in this example was tested with the 5% change, the change did not resonate with any category. Thus, the system performed as expected in this case. The system also did not resonate with a 4% increase, but did resonate with a 3% increase and below.

To investigate the theory, the single-point classification region shown in Figure 4-49 was tested with a 5% change. The actual indigivallance that was used in the training of the neural net was 0.976. From Equation (47), it is seen that the threshold for change is

$$\begin{aligned}\Delta &> M - Mp^p \\ \Delta &> 2 - 2 \cdot 0.976 \\ \Delta &> 0.048\end{aligned}$$

The single-point classification region was tested with a change of 5%, 4.79%, and 4.81% to investigate the validity of the theory. The results are shown in Table 4-2.

As predicted, the threshold of change detection documented in Table 4-2 was at or very near the change equaling 4.8% of full scale, thus verifying the theory. The 0.976 vigilance allowed a 5% change to be detected when a single classification region was being investigated.

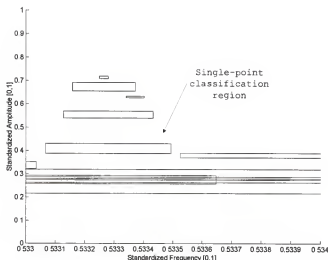


Figure 4-49. 11th to 16th Blade set classification regions.

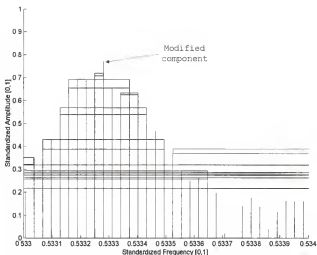


Figure 4-50. 11th to 16th Blade set with 5% change at standardized frequency of approximately 0.53328.

Table 4-2. Results of single-point classification-region tests.

Change (Δ)	Change Detected?
4.79%	No
4.81%	Yes
5%	Yes

The J Matrix

The **J** matrix of Equation (49) creates a metric in a set of new dimensions that allows placement of vibration information into different sections of the input space. The metric separates the input space into regions such that an input vector in one region will not

resonate with a node containing information from another region. The concept of resonance fields was developed to determine the distance between each point in the **J** matrix. The **J** matrix was used to solve problems in the detection of changes where changes in spectrum components would not be detected. Changes would not be detected if the changed input vector resonates with a different node than originally programmed the component. Ideally, only changes in amplitude would be detected but Fuzzy ART detects changes in all dimensions. If an input vector, that the neural network had learned at a certain amplitude and frequency, increased in amplitude sufficiently to not resonate with the previous, learned prototype it should be deemed novel. Due to the close spacing of the features in the frequency dimension, the new input vector may resonate with a nearby feature, constituting a false positive classification. The use of the metric provided by the **J** matrix allows separation of neighboring features in the input space.

Using knowledge of the features in the turbine spectrum, input vectors representing different features were separated using different values in the **J** matrix. An example of a two-dimensional **J** matrix is shown in Figure 4-51. The **J** values contained in the input vector are constrained to occur at the discrete points shown in the figure. As will be described, the points in the **J** matrix of Figure 4-51 allow enough additional sufficient ℓ^1 -norm separation in the input space to ensure that vectors augmented with one point in the **J** matrix will not resonate with prototypes learned from vectors that had been augmented with a different **J** matrix point.

The **J** matrix in Figure 4-51 contained 361 different regions. It contains the maximum amount of regions possible to separate resonance fields associated with detecting a 5% change, as will be explained in

this section. Fewer regions could be used, but not more. The use of more regions would not guarantee separation. In some of the testing, 869 different feature separations were desired. A new dimension had to be added to the **J** matrix to represent these regions. Figure 4-52 shows the points in the three-dimensional **J** matrix that will separate the resonance fields. Using three dimensions provides 6859 different regions in the input space.

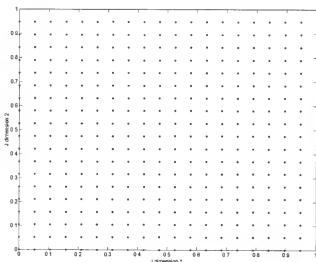


Figure 4-51. Locations in a 2-D **J**-matrix to separate resonance fields associated with 5% change detection.

The **J** matrix codes a priori information into the input signal. This information is derived from the turbine tachometer frequencies and knowledge of the mechanical construction of the turbine rotating elements. Feature separation allows the Fuzzy ART neural network to spread out the learned categories through higher-dimensional space,

instead of just two dimensions, creating greater distances between categories.

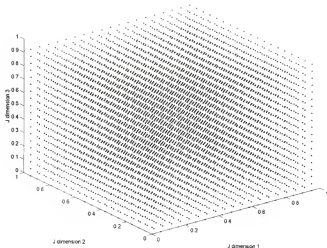


Figure 4-52. Locations in a 3-D J-matrix to separate resonance fields associated with 5% change detection.

The increased resolution provided by the additional dimensions is needed to separate features that are close together. The information in the amplitude dimension represents approximately 40 dB and a maximum 5% change is being detected, but the frequency dimension extends across 20kHz, with some features only taking up 4 or 5 Hz, a 0.02% change. The regions in which a signal can be classified as a certain category will be referred to as the resonance field. Resonance fields extend out from the learned templates and can overlap. When an amplitude component related to a feature is learned, it creates a template with a resonance field. As the signal changes, if the amplitude remains

within the resonance field it will have direct access to the learned template. Direct access means that the node with the highest activation, $T_j(I^P)$, also has the highest resonance. In direct access, the node is not reset and no vigilance search of the remaining nodes is required.

If the signal changes sufficiently to cause its amplitude to leave the resonance field of the learned template, then that case will not have direct access to the learned template that used to represent this feature. Ideally, that would be sufficient to cause the Fuzzy ART to not recognize the feature and report it as novel, sending the information to the trending system, but there is a problem.

After the case fails to resonate with the learned template that it used to resonate with, a vigilance search is performed that checks the other long-term memory nodes for an acceptable match. These nodes will be checked in order of decreasing amplitude until a node is found that resonates, or no other node resonates. If no other node resonates, then the case is deemed novel. If the amplitude of the changed signal falls into the resonance field of a different node, then that node will categorize the signal, and it would not be deemed novel.

Having another node categorize this signal can be undesirable if that node was created to recognize a different spectral feature than the feature being considered. Due to the close spacing of the frequency features in the standardized signal, overlapping of resonance fields is common. To minimize problems with overlapping resonance fields, extra distance is provided between features by preclassifying, or separating, those features using the a priori information. If enough distance is provided to ensure that the resonance fields of each known feature cannot overlap with the resonance fields of other

features, then the use of this neural network becomes much more robust for this application.

The information used for a priori feature separation will be presented. The theory behind resonance fields that was developed will be introduced, along with an illustration of the problem being solved. The creation of the **J** matrix will be described. The neural network with the expanded dimensionality will be trained and tested to show the efficacy of this method.

Turbine vibration characteristics: a priori information

Primary spectral features in a turbine include energy created by the once-per-revolution vibration, turbine blade-sets, combustion noise, exhaust rumble, gear noise and harmonics of the rotating frequencies with all other features. In the LM2500, other known features occur when there are problems developing in the turbine. These include rotating stall and oil slosh. Rotating stall occurs when some of the airflow in the blade sets is disrupted and blades go into aerodynamic stall. This stall rotates axially around the turbine blades, creating a spectral feature related to this rotation.

Oil slosh occurs when oil seeps through the oil-bearing seals into the interior of the rotor case. Oil slosh is distinguished from other oil vibration effects such as oil whip and oil whirl, which occur in the oil filled bearing itself, not the interior of the rotor. This trapped oil is forced into rotation because it is interior to the spinning rotor case, causing severe vibration due to the imbalance caused by the mass of the oil. The oil rotates at a fraction of the rotor speed, creating what is termed a sub-synchronous vibration.

Resonance fields

Each long-term memory template stored in the F_2 layer of Fuzzy ART has a resonance field associated with it. The size of the resonance field is determined by the vigilance, ρ , that is being used to reference that long-term memory template and by the size of the classification region. The use of fuzzy operators carves the input space into different regions that are more complicated than would occur with the use of l^1 - or l^2 norms to indicate proximity of the input vector to the weight vector in the node. The complication arises because Fuzzy ART compares the input vector to a vector range stored in the weights of each node; the ranges being stored using complement-coding.

A limited demonstration of the effects of this field was shown in Table 4-2, where signal changes below a certain amount resonated with the learned template. Those changes, greater than the threshold, did not resonate and were deemed to be changes that were detected.

As was seen in Figure 4-50, a spectral feature may consist of multiple spectral components, including a peak and the sides of the main lobe of the feature. If the increased spectral component was not the peak of a major spectral feature, then a positive-going change may resonate with another classification region representing higher amplitude components of that feature. If the increased spectral component were the peak, it would be expected that no other classification region would be able to resonate with it if its amplitude increased. This is not always the case, due to false positive classification by neighboring resonance fields.

From Equation (47), the minimum amount, Δ , that a signal must change in order to guarantee that it will not resonate with the node

trained to recognize the unchanged signal can be determined. From Equation (47),

$$\frac{M-\Delta}{M} < \rho^p$$

thus,

$$\Delta > M(1-\rho^p) \quad (63)$$

to ensure that the original node will not resonate with the changed signal.

In a one-dimensional case, the stored template holds the pattern $w=(u,v^c)$, where u and v are scalar values, and $v=1-u$. The input case scalar a yields the input pattern $I=(a,a^c)$. For the one-dimensional case, Δ can be determined from Equations (41) and (47), and by setting ρ to the boundary condition $\rho=(M-\Delta)/M$:

$$\begin{aligned} \frac{\tilde{a}_1 \wedge \tilde{w}_1 + \tilde{a}_1^c \wedge \tilde{w}_2}{\tilde{a}_1 + \tilde{a}_1^c} &< \rho^p \\ \frac{a \wedge u + a^c \wedge v^c}{a + a^c} &< \rho \\ a \wedge u + a^c \wedge v^c &< \frac{M-\Delta}{M} \\ a \wedge u + a^c \wedge v^c &< (1-\Delta) \\ 1 - a \wedge u - a^c \wedge v^c &> \Delta \end{aligned} \quad (64)$$

There are two extremes of the resonance field possible in the one-dimensional case. These occur when $a < u$ or $a > v$. In the case where $a < u$, Equation (64) becomes

$$\begin{aligned} 1 - a \wedge u - a^c \wedge v^c &> \Delta \\ 1 - a - v^c &> \Delta \\ 1 - a - (1-v) &> \Delta \\ v - a &> \Delta \end{aligned} \quad (65)$$

If $a > v$, then Equation (64) becomes:

$$\begin{aligned}
 1 - a \wedge u - a^c \wedge v^c &> \Delta \\
 1 - u - a^c &> \Delta \\
 1 - u - (1 - a) &> \Delta \\
 a - u &> \Delta
 \end{aligned}
 \tag{66}$$

Equations (65) and (66) are depicted in Figure 4-53 for the case where the learned template, w , consists of a single point. It is seen that the resonance field extends a distance Δ on either side of the single point that had been learned in long-term memory. If a new case is encountered that consists of a point that falls within the resonance field, Q , and the node containing this single-point category has the highest activation, then the new case will resonate with this node.

In the one-dimensional case, if the long-term memory has learned more than one point, then its template will have the form of a line segment. In the template $w = (u, v^c)$ the scalar variable u represents the lower valued point and v (not v^c) represents the higher valued point. The extent of the resonance field is still defined by Equations (65) and (66). There is a difference in the resonance field from the single-point case, though. Figure 4-54 shows the resonance field for a template that has coded multiple different points.

The size of the resonance field decreases as the pattern grows in the long-term memory node. As the network learns, it converges to categories of the size governed by the vigilance. This characteristic performs an ambiguity reduction function. In the limiting case, the maximum size that the 1-D template can grow to is a line segment of length Δ , as shown in Figure 4-55.

In the two-dimensional case, the resonance field supports changes throughout both dimensions. It exhibits similar characteristics to the one-dimensional resonance field in that the resonance field shrinks as

the template grows, but provides more complexity in the shape of the field. A two-dimensional long-term memory template has the form

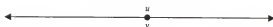
$$w = (u_1, u_2, v_1^c, v_2^c).$$

The two-dimensional input case and complement-coded input vector are:

$$a = (a_1, a_2)$$

$$I = (a_1, a_2, a_1^c, a_2^c).$$

Note: $v = w$ in a single point template.



Template contains a single 1-D point.



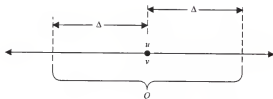
case: $a' > v$

Point a' resonates with template containing single point u .



case: $a' < u$

Point a' does not resonate with template containing single point u .



Resonance field, Q , for 1-D template containing one point.

Figure 4-53. Resonance field for 1-D template containing 1 point.

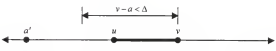


Template (dark line segment) has learned multiple 1-D points.



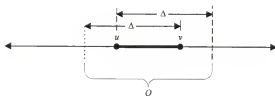
case: $a' > v$

Point a' resonates with multipoint template.



case: $a' < u$

Point a' does not resonate with multipoint template.



Resonance field, Q , for 1-D multipoint template.

Figure 4-54. Multipoint template resonance field.



Figure 4-55. Limiting case: template expands to the size of the detectable change threshold.

The resonance field is calculated from the vigilance inequality, beginning as follows:

$$\begin{aligned}
 \frac{\| \mathbf{l} \wedge \mathbf{w} \|}{\| \mathbf{l} \|} &< \rho \\
 \frac{a_1 \wedge u_1 + a_2 \wedge u_2 + a_1^c \wedge v_1^c + a_2^c \wedge v_2^c}{a_1 + a_2 + a_1^c + a_2^c} &< \rho \\
 \frac{a_1 \wedge u_1 + a_2 \wedge u_2 + a_1^c \wedge v_1^c + a_2^c \wedge v_2^c}{2} &< \frac{M - \Delta}{M} \\
 \frac{a_1 \wedge u_1 + a_2 \wedge u_2 + a_1^c \wedge v_1^c + a_2^c \wedge v_2^c}{2} &< \frac{2 - \Delta}{2} \\
 2 - a_1 \wedge u_1 - a_2 \wedge u_2 - a_1^c \wedge v_1^c - a_2^c \wedge v_2^c &> \Delta
 \end{aligned} \tag{67}$$

Although the F_2 template can consist of just a single point, in general it will have grouped several points into its classification region. In the case of a single-point classification region, the decision boundary at the edge of the resonance field is described as the locus of points having a constant L1 norm value of Δ away from the single point. After the classification region has grown by classifying more points, it can be represented as a rectangle, as shown in Figure 4-40. The width of the rectangle is represented in Figure 4-56 as δ_1 and the height is shown as δ_2 . The decision boundary of the resonance field is no longer represented by the L1 norm, but is governed by its position, or region, outside the classification region. If the region is in an area perpendicular to a vertical or horizontal edge of the classification region, then the decision boundary is at a distance $\Delta - \sum_{i=1}^M \delta_i$ from the edge of the classification region, where δ is the length of an edge in one dimension of the classification region and M is the number of dimensions in the input space. At each corner of the classification region, the decision boundary is again represented by a constant L1

norm in the region diagonal to the corner, such as in region II shown in

Figure 4-56. The constant L1 norm length is again $\Delta - \sum_{i=1}^M \delta_i$.

There are eight possible cases for the relationship between the input cases and the stored template, corresponding to the eight divisions of the 2-D space shown as in Figure 4-56.

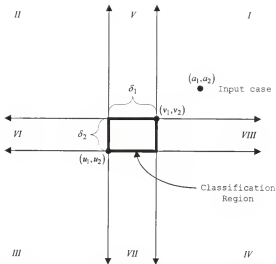


Figure 4-56. Analysis sections for 2-D resonance fields.

In Section I of Figure 4-56, the horizontal dimension, a_1 , of an input case is greater than that of both the lower left (u_1) and upper right (v_1) corners of the long-term memory template. Likewise, $a_2 > u_2$ and $a_2 > v_2$. The resonance field for this section can be calculated from Equation (67) as

$$\begin{aligned}
2 - a_1 \wedge u_1 - a_2 \wedge u_2 - a_1^c \wedge v_1^c - a_2^c \wedge v_2^c &> \Delta \\
2 - u_1 - u_2 - a_1^c - a_2^c &> \Delta \\
2 - u_1 - u_2 - (1 - a_1) - (1 - a_2) &> \Delta \\
2 - u_1 - u_2 - 1 + a_1 - 1 + a_2 &> \Delta \\
a_1 + a_2 - u_1 - u_2 &> \Delta \\
(a_1 - u_1) + (a_2 - u_2) &> \Delta
\end{aligned}
\tag{68}$$

Equation (68) defines a region in 2-D space. Where $(a_1 - u_1) + (a_2 - u_2) > \Delta$, the input case will not resonate with the node. The resonance field calculations for the other sections of Figure 4-56 are shown in Table 4-3.

Table 4-3. Resonance field component calculations.

Section	Horizontal Case	Vertical Case	Governing Equation
<i>I</i>	$a_1 > u_1$ $a_1 > v_1$	$a_2 > u_2$ $a_2 > v_2$	$(a_1 - u_1) + (a_2 - u_2) > \Delta$
<i>II</i>	$a_1 < u_1$ $a_1 < v_1$	$a_2 > u_2$ $a_2 > v_2$	$(v_1 - a_1) + (a_2 - u_2) > \Delta$
<i>III</i>	$a_1 < u_1$ $a_1 < v_1$	$a_2 < u_2$ $a_2 < v_2$	$(v_1 - a_1) + (v_2 - a_2) > \Delta$
<i>IV</i>	$a_1 > u_1$ $a_1 > v_1$	$a_2 < u_2$ $a_2 < v_2$	$(a_1 - u_1) + (v_2 - a_2) > \Delta$
<i>V</i>	$a_1 > u_1$ $a_1 < v_1$	$a_2 > u_2$ $a_2 > v_2$	$(v_1 - u_1) + (a_2 - u_2) > \Delta$
<i>VI</i>	$a_1 < u_1$ $a_1 < v_1$	$a_2 > u_2$ $a_2 < v_2$	$(v_1 - a_1) + (v_2 - u_2) > \Delta$
<i>VII</i>	$a_1 > u_1$ $a_1 < v_1$	$a_2 < u_2$ $a_2 < v_2$	$(v_1 - u_1) + (v_2 - a_2) > \Delta$
<i>VIII</i>	$a_1 > u_1$ $a_1 > v_1$	$a_2 > u_2$ $a_2 < v_2$	$(a_1 - u_1) + (v_2 - u_2) > \Delta$

When the governing equations are set to equalities, the line separating the resonance field from the rest of the template space can be plotted. In each of the sections, a different line is constructed, based upon the governing equation. Figure 4-57 shows an example resonance field for a rectangular classification region.

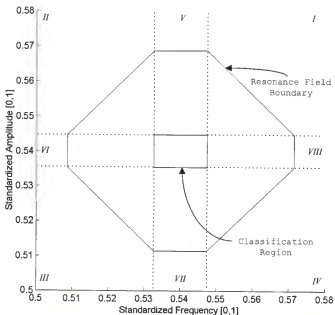


Figure 4-57. Example 2-D Resonance Field.

The meaning of the resonance field will be illustrated with respect to input vector information. If a vector falls within the resonance field, then it will program the node. The prototype, or classification region, contained in the node will adapt to contain the new vector. If the input vector falls outside the resonance field, it

will be deemed novel and the node will not adapt to learn the new vector. Figure 4-58 shows two vectors in the neighborhood of a 2-D resonance field. Vector **A** falls within the resonance field, but vector **B** extends beyond it. The classification region adapts to encompass **A**, as shown in Figure 4-59.

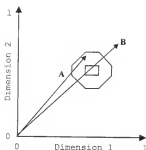


Figure 4-58. Two vectors near resonance field.

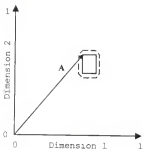


Figure 4-59. Response of network to vector **A**.

The resonance field also adjusts because more of the classification region has been consumed, yielding less uncommitted area in the classification region to adapt to new vectors. Vector **B**, shown

in Figure 4-60, does not resonate with node and causes no changes to the classification region or resonance field.

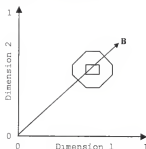


Figure 4-60. Response of network to vector **B**.

The size of the resonance field shrinks as the size of the classification region grows. Referring to Figure 4-56, the size of the 2-D classification region is $\delta_1 + \delta_2$. The maximum size of the 2-D classification region occurs when $\delta_1 + \delta_2 = \Delta$, in which case the resonance field shrinks to the same size and orientation as the classification region. The resonance field shown in Figure 4-56 is the most general form of the resonance field, that pertaining to a rectangular template. If the template contained a single point, the resonance field changes to a diamond. If the template contains a horizontal line segment, the vertical components of the resonance field (sections VI and VIII) disappear. For a vertical line segment, the horizontal resonance-field components (sections V and VII) disappear.

To test the theory concerning the resonance fields, a million test cases were run through the Fuzzy ART neural net to plot the actual region. A single classification region from the turbine data set was selected and a dense grid of input cases was created to cover the

expected resonance field. The results of this test are shown in Figure 4-61.

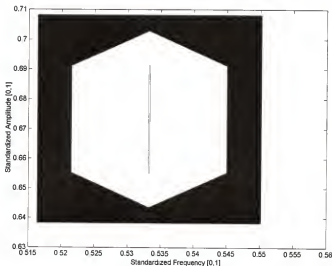


Figure 4-61. Resonance field test for single classification region.

As seen in Figure 4-61, the resonance field theory performed as expected. All the points shown in black were outside the resonance field and did not resonate with the single node in the neural net.

Using actual turbine data, the high-vigilance classification regions of Figure 4-47 were analyzed to determine the resonance fields. The results of this analysis are shown in Figure 4-62.

The overlap of resonance fields can be seen in Figure 4-62. This overlap presents a problem to detecting changes in spectral components. If the spectral component amplitude change is great enough to leave the resonance field of the node that had previously classified it, it is desired that the system detect this as a novel case, a trending change.

There are circumstances where the spectral component may leave one resonance field and enter the resonance field of another feature. After a vigilance search, the changed component would resonate with that node. This is undesirable. To avoid this problem it is necessary to separate the resonance fields of the two features by at least the horizontal distance Δ . The maximum distance that the resonance field can extend from a classification region is Δ , thus a separation greater than Δ ensures that the vertical components of two features will not enter each other's resonance fields.

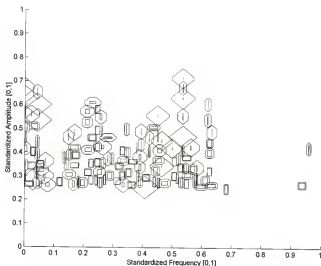


Figure 4-62. Resonance fields for high-vigilance turbine data.

An example of the problem arising from close resonance fields is shown in Figure 4-63. The spectral feature with the greatest amplitude exists at a standardized frequency of approximately 0.533. This feature consists of multiple components grouped together to provide the

sloped edges of the feature. One of the components has a wide resonance field because its classification region consists of a single point. Another feature at standardized frequency of approximately 0.561 has a resonance field at its peak that overlaps the resonance field of the component of the other feature. If the second feature amplitude grows sufficiently to leave its learned resonance field and enter the resonance field of the other feature, then the change will not be detected.

Although the change shown in Figure 4-63 was more than the 5% change that was theoretically detectable, it was not detected because it fell into the resonance field of another feature. A closer view of the two resonance fields of interest is shown in Figure 4-64.

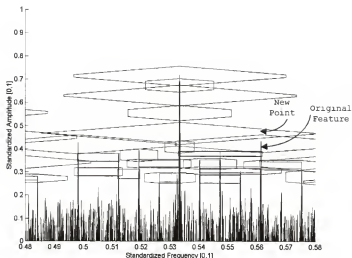


Figure 4-63. Component analysis identification.

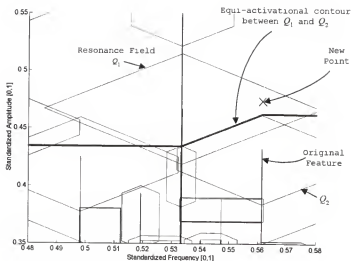


Figure 4-64. Illustration of problem of extended resonance fields.

The Q_1 resonance field in Figure 4-64 extends over the top of the Q_2 resonance field, causing mistakes in the processing of the original feature as it changes amplitude. As the original feature increases in amplitude to the level of the new point shown, it does fall out of the Q_2 resonance field, which should cause the neural net to indicate that a novel change has occurred. Instead, the feature crosses into the resonance field Q_1 . To ensure that the changed feature will actually be chosen as resonating with the Q_1 resonance field, the equi-activational contour that exists between the Q_1 and Q_2 resonance fields was plotted. The equi-activational contour is the line where the activations corresponding to the two classification regions are equal in magnitude. Crossing the equi-activational contour into the Q_2

resonance field ensures that the activation of the template associated with Q_2 is greater than the activation associated with Q_1 . Thus, the new point directly accesses the category associated with the Q_2 resonance field. The changed feature is not recognized until its amplitude increases beyond the Q_2 resonance field.

To provide more separation between the features, a dynamic classification of the features, using a priori information, is used to augment the classification space with additional dimensions supplying sufficient ℓ^2 -norm separation between the resonance fields. The classifications are based upon turbine construction and the instantaneous turbine rotational velocities. Features that are known to exist in the spectrum are each given a different position in four-dimensional space. The additional dimensionality is chosen to ensure that there will be at least a Δ difference between each known feature, thus providing effective separation.

The input matrix, A , takes on the form presented in Equation (49), reproduced below:

$$A = \begin{bmatrix} F(1) & V(1) & J(1,1) & J(1,2) \\ \vdots & \vdots & \vdots & \vdots \\ F(k) & V(k) & J(k,1) & J(k,2) \\ \vdots & \vdots & \vdots & \vdots \\ F(K) & V(K) & J(K,1) & J(K,2) \end{bmatrix}$$

The J matrix columns provide the added dimensionality. Fifty-one features were identified in the turbine spectrum. The remaining input vector information retains its original location in the J dimension. It is desired to insert a distance of at least Δ into the dimensionality between each feature. In binary neural networks this additional dimensionality could be accomplished by 1-of- C coding, where each identified feature would have a new dimension created that indicates

feature presence with a value of one or zero. Because Fuzzy ART uses analog inputs, it can represent the needed dimensionality shifts using analog variances in fewer dimensions. To represent a shift in four-dimensional value of at least $\Delta=0.05$, so a value of $1/19 \approx 0.05263$ was chosen to offset each feature. The $J(2)$ position in each the J matrix will hold the steadily increasing dimensional shift. The $J(1)$ position will increment by $1/19$ each time the $J(2)$ position passes its maximum value of $1-(1/19)$. For the fifty-one features, the J values that code these features are

$$\begin{aligned} J(2) &= j \bmod(19), & 1 \leq j \leq 51 \\ J(1) &= \left\{ \left(\frac{j}{19} \right) - \left[\left(\frac{j}{19} \right) \bmod(19) \right] \right\} \bmod(19), & 1 \leq j \leq 51 \end{aligned} \quad (69)$$

where

j = feature number.

The features that were used for the LM2500 turbine were outlined in the previous section. This technique could be called Fractional-Dimension 1-of-C coding.

A spread of four component bins around each feature was used to ensure that the entire feature would be contained in the adjusted dimensionality. The feature classification was based upon priority. Some known features are wider than others, but higher priority features will overwrite the classification of lower priority features. So, known features were grouped into one a priori feature, and given the highest priority.

Within each dimensionally-adjusted feature space, the resonance field can grow to the maximum without affecting any other dimensionally-adjusted feature. An illustration of the use of this feature separation technique is provided in Figure 4-4.

Training and testing with expanded input dimensionality

The neural network was trained with the expanded input dimensionality and a priori feature separations. The resulting system was tested with the same point that failed to be detected with the 2-D system. The point could then be detected. The 4-D neural network required five epochs to train, taking 40 seconds. There were 163 long-term memory (F_2) nodes committed.

The resulting resonance fields are shown in Figure 4-65. To visualize the new resonance fields for the 4-D inputs, the first two dimensions and their associated complements, were used to create Figure 4-65. The neural dimensions associated with the a priori classifications were ignored for the printout. Thus, all the resonance fields that were created are shown in a single 2-D plane.

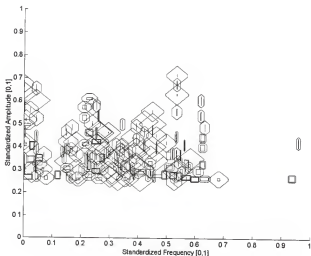


Figure 4-65. All resonance fields and classification regions from 4-D analysis, shadowed onto a 2-D plane.

The features that were analyzed using the 2-D neural network, seen in Figure 4-63 and Figure 4-64, are examined using the results of the 4-D training. The feature that was changed in the 2-D analysis, at standardized frequency of approximately 0.562, was not related to any a priori classification. Therefore, it fell into the leftover spectrum information, termed Feature 0. The Feature 0 plane is shown in Figure 4-66. The feature that had been the problem in the 2-D analysis, the 11th through 16th blade set feature, was known and classified into its own four-dimensional position. All the resonance fields and classification regions associated with the 11th through 16th blade set four-dimensional plane are shown in Figure 4-67. The ability to differentiate features from the remaining spectral information is apparent in Figure 4-67, because only the information related to this feature are shown there. A close-up of Figure 4-66 is presented in Figure 4-68, where it is seen that the feature at a standardized frequency of approximately 0.562 does not have the interfering resonance field above it. Changes in this feature were thus detected.

One of the advanced spectral analysis and monitoring techniques used in practice involves the use of acceptance envelopes. These acceptance envelopes are defined in the spectrum and provide amplitude limits for various features in the spectrum. These acceptance envelopes previously were defined by vibration analyst experts. The use of a neural network to generate these acceptance envelopes, or classification regions, increases the obtainable sensitivity to change. The new system will also detect and track unexpected features in the waveform. Because the analysis is primarily performed in software, the system could be made small enough and inexpensive enough to include with installed turbines as a permanent condition monitor.

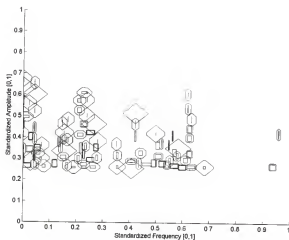


Figure 4-66. Resonance fields and classification regions for Feature 0.

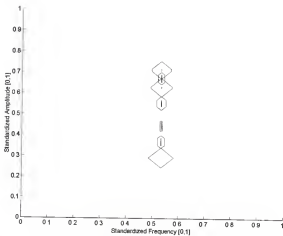


Figure 4-67. Feature dimension for turbine blade set 11-16, resonance fields and classification regions.

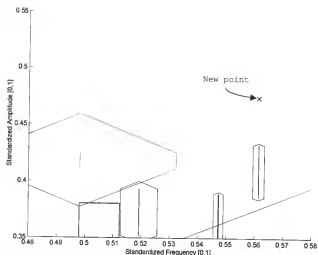


Figure 4-68. Close-up of resonance fields and classification regions of Feature 0, showing non-interference to feature at standardized frequency of approximately 0.562.

The use of a priori classifications provides more reliable detection characteristics in this approach to detecting trends in spectrum components. The fact that the neural network learns the information in the spectrum and can monitor the entire spectrum for changes thus provides capabilities that did not previously exist. The input spectrum has a large number of points, 32768 components, but the neural network compresses this information to only 163 classification regions in this case. This enhances the opportunity to implement the system, because of reduced memory requirements. A large reduction in the number of classification regions was achieved using individual case vigilance. Individual case vigilance is a technique that has been suggested in the literature [33], but apparently not developed in the papers and books. The use of individual case vigilance requires a

small but obvious change in the standard Fuzzy ART network and training. The use of this neural network technique to provide automated narrow band spectral envelope construction was less obvious. The alacrity of classification and precision obtainable using this technique may benefit the field of condition based monitoring.

Multiple-Spectrum Training Sets

In the discussion of the theory behind the development of the neural network inputs, a single spectrum was used for training. While this permitted repeatable results throughout the analysis, in actual application the network would be trained with multiple spectrums to automatically provide a statistical baseline spectrum. Each spectrum in the multiple-spectrum training set would be obtained by performing a Welch method variance-reduction algorithm on the input data.

The concatenation of multiple spectrums into a training set occurs after variance reduction techniques have been performed and the resulting spectrum has been standardized and preprocessed. The statistical baseline spectrum classifications would be programmed into the system for application and modified as required to reflect the actual installation characteristics. The neural classification prototypes should become more statistically valid when multiple spectrums are learned instead of just one spectrum. The normal variance of the turbine operation will thus modify and fill out the prototypes. This should result in a more stable system, with fewer new neurons being committed in a given duration.

To investigate the use of multiple spectrums, recordings of the turbine vibration at approximately the same operating point were used as input to the neural network. Referencing the Figure 2-5, LM2500

Tape 2 Forward, Gas Generator spectrogram, it is seen that the turbine operating point stays steady for approximately 300 seconds. The data used in the analysis to this point was obtained from this same area. Multiple spectrums were obtained from this operating point and were tested using the spectrum that had been used for analysis in the development of the neural net input preprocessing. Five spectrums, of 32768 points were used for training. They were concatenated into a 163840 case training set.

The two-dimensional shadow of all of the four-dimensional resonance fields and classification regions for the neural net trained with multiple datasets is shown in Figure 4-69. There are more classification regions, or nodes, dedicated when training with the more realistic situation of multiple data sets. There were 331 long-term memory nodes committed, requiring 32 epochs to learn, with a learning time of 46 minutes and 25 seconds. The programming time was more extended, likely due to the number of patterns that needed to be repetitively examined. The network compressed the information in the 163840 case training set, with each case containing five floating points (including case vigilance) into only 331 nodes, each containing eight floating point values. Thus, 2648 floating-point values were needed to represent the 819200 floating point values in the training set.

The resonance fields and classification regions for feature set 0 are shown in Figure 4-70 and for the 11th to 16th blade set feature in Figure 4-71. These graphs show the higher level of detail provided by more points in the training set. Comparison of the 11th to 16th blade set feature for one spectrum training (Figure 4-67) with that of multiple spectrum training (Figure 4-70) shows that more details are apparent in the multiple spectrum case, but the feature has the same

basic vertical shape. There were seven classification regions in the single spectrum case and twelve in the multispectrum case. This is not a large increase for five times the amount of training data.

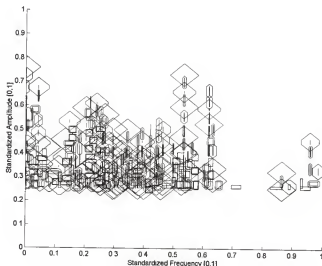


Figure 4-69. 2-D shadow of all 4-D resonance fields and classification regions for multiple data sets.

Inspection of Figure 4-72 shows that there is no resonance field extending over the top of the feature classification existing at the standardized frequency of approximately 0.562. Thus, the change in this feature was still detected in the multispectrum training data case. This shows that the extension of this method to a more real world situation retains stability as more information is learned and that this feature is a relatively stable feature in the spectrum. If this feature changes with engine wear, the system will detect the change.

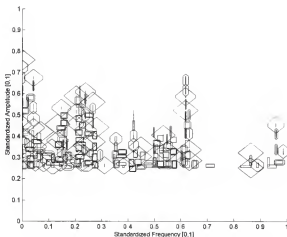


Figure 4-70. Resonance fields and classification regions for Feature 0, for multiple data sets.

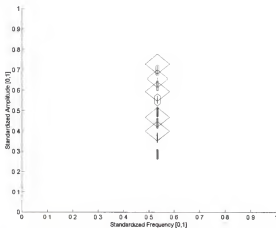


Figure 4-71. Feature dimension for turbine blade set 11-16, resonance fields and classification regions for multiple data sets.

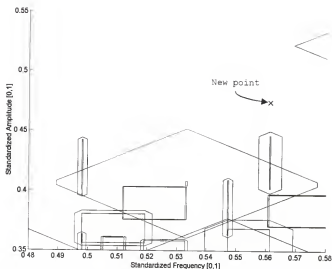


Figure 4-72. Close-up of resonance fields and classification regions of Feature 0, showing non-interference to feature at standardized frequency of approximately 0.562 for multiple data sets.

LM6000 Turbine Application

The bulk of the neural network analysis concerned the LM2500 gas turbine because most detailed information was available for it and the recorded tapes showed operation over a wide dynamic range. The LM6000 gas turbine recorded for this research was used in a power generation capacity, where it was run continuously at a nearly constant speed. The LM6000 would provide an excellent application for the Fuzzy ART Condition Based Maintenance system because the constant speed operation would require fewer neural knowledge sets to be created for the different modes. The LM6000 knowledge set would contain prototype categories for different times of day and different seasons. These

knowledge sets would then be selected for use by the mode detection subsystem. To apply the neural CBM system to the LM6000 would require access to detailed information concerning the blade-stages and other known features in the engine.

The recorded LM6000 signals were preprocessed and analyzed by the Fuzzy ART neural network in the same manner as the LM2500 signals, with the only exception being a different choice of v_{min} for the dynamic case vigilance algorithm. The unprocessed LM6000 spectrum is shown in Figure 4-73. The sensor calibration from the LM2500 analysis was applied to this data, because the LM6000 sensor data sheet was not obtained, but the decibel representation in this figure is valid. The signal was standardized and classified by the Fuzzy ART network. The results of this classification are shown in Figure 4-74 and Figure 4-75. The recorded LM6000 signals tended to have a lower noise baseline than the LM2500 signals, so v_{min} was adjusted to a lower value. The pseudo-CDF of the LM6000 signal, shown in Figure 4-76, was used to find the v_{min} level in the same manner as the LM2500 signal data.

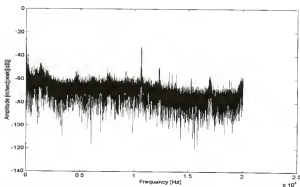


Figure 4-73. LM6000 spectrum, calibrated per LM2500 sensor.

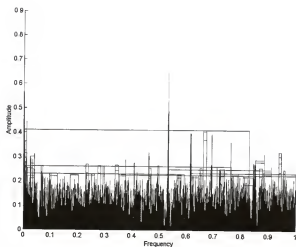


Figure 4-74. Fuzzy ART classification of LM6000 standardized spectrum.

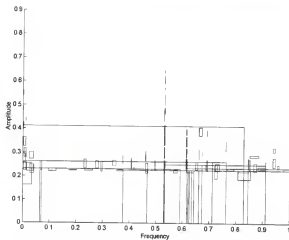


Figure 4-75. Fuzzy ART classification regions for LM6000 standardized spectrum.

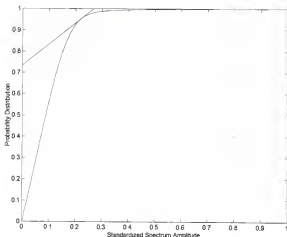


Figure 4-76. Pseudo-CDF of LM6000 spectrum.

Testing

The testing regimen that has been applied to the vibration analysis case involves applying a series of changes to the spectrum information and checking how many novel cases are generated. Ideally, all changes should be detected. To create the test set, the spectrum that the system saw trained with was modified to insert a 5% change in every frequency component. For spectrum components having amplitudes below the noise baseline, the amplitude was changed to equal the noise baseline plus 5% and their case vigilance settings adjusted to match the case vigilance settings of the other higher amplitude components. Each new frequency component is a single case and there were 32768 components. A perfect result would be 32768 changes detected.

In each of the test results to follow, the number of epochs required to train, the number of committed nodes and the training time

are given in the table caption. The 4-D case with 51 a priori classifications was tested with the following results:

Table 4-4. 4-D testing results, 51 a priori classes.
Epochs: 4, nodes: 215, time 62 sec.

Change amount Δ	# changes detected	Percentage of expected detections
5%	18925	57.8%
6%	20154	61.5%
7%	21274	64.9%
8%	21365	65.2%

The results in shown in Table 4-4 show that many of the changes were not detected. The low performance shown in the 5% column of Table 4-4 indicates a large percentage of false positive classifications ($100\% - 57.7\% = 42.2\%$ false positive classifications). This was due to proximity of changed data to previously learned features. A plot of the detected changes appears in Figure 4-77. Changes that were detected are drawn. Undetected changes are blank in the figure. Apparently, most of the major features are detected, but there are some features missing and many blank sections.

To supply more dimensionality shift between the spectral components, more a priori classification regions were used. All the harmonics of the gas generator and the first ten harmonics of the power turbine 1X speeds were added as a priori classifications. This increased the number of a priori classifications to 201. To ensure that enough dimensionality shifts could be supplied, another vector was added to the J matrix, allowing up to 6859 individual 5% dimensionality-shifts. The 5-D network was tested with the results shown in Table 4-5.

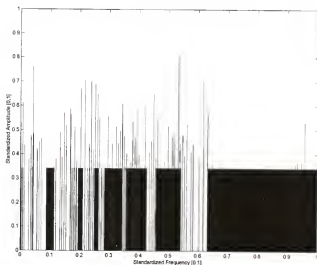


Figure 4-77. Spectrum changes detected by 4-D neural net.

Table 4-5. 5-D testing results, 201 a priori classes.
Epochs: 4, nodes: 215, time: 57 sec.

Change amount Δ	# changes detected	Percentage of expected detections
5%	27926	85.2%
9%	24632	78.2%

Using the additional a priori classifications increased the number of changes detected. Interestingly, larger changes were detected less than small changes here, likely due to resonance with classification regions nearby in frequency.

To help separate frequency regions, the entire frequency dimension was chopped up into 40 ranges, each containing a frequency range of 500 Hz, or 2.5% of the entire spectrum. Each small section

was given its own dimensional shift. This increased the number of a priori classifications to 241. The results were as follows:

Table 4-6. 5-D testing results, 241 a priori classes.
Epochs: 4, nodes: 215, time: 54 sec.

Change amount Δ	# changes detected	Percentage of expected detections
5%	26638	81.3%
9%	28906	88.2%

To try to increase the number of classifications, instead of training with the noise baseline data at a lower case vigilance, the data points having amplitudes below the noise baseline threshold were eliminated. Only frequency components having amplitudes greater than the noise threshold, v_{min} , shown in the discussion of Figure 4-45 were allowed to reach the neural network for training. Using the same 500 Hz chopping length for the frequency spectrum and only training on the signal higher than v_{min} yielded the following performance:

Table 4-7. 5-D testing results, 241 a priori classes, no noise.
Epochs: 4, nodes: 215, time: 55 sec.

Change amount Δ	# changes detected	Percentage of expected detections
5%	25755	78.6%
9%	28996	88.5%

Deleting the noise information did not provide an improvement, but also did not greatly decrease the performance. The only benefit would be that of less raw data to be stored in a section of the algorithm.

Chopping the frequency spectrum into 40 regions provided separated frequency changes more than 2.5% of the frequency range, but because of the large amount of spectrum covered (~20kHz), multiple

features could be contained in this range. The minimum spacing between gas generator harmonics is 50 Hz, because the minimum operating speed of the gas generator is 50 Hz. To ensure that frequency ranges of less than 50 Hz are separated, a chopping length of 30 Hz was chosen. At each 30 Hz increment, the input cases are dimensionally shifted by 5%, thus more providing separation between nearby components. There were 668 frequency-chopping classifications created. Along with the existing a priori classifications of blade pass, oil slosh, etc, this totaled 869 a priori classifications. Again, the frequency-chopping classifications do not require detailed subject matter expertise to implement. The added a priori classifications provided a substantial improvement, as shown in the following table:

Table 4-8. 5-D testing results, 869 a priori classes.
Epochs: 5, nodes: 236, time: 76 sec.

Change amount Δ	# changes detected	Percentage of expected detections
5%	31540	96.3%
9%	32195	98.3%

The performance with 869 a priori classifications was also checked using only the frequency components exceeding v_{min} , i.e. components having amplitudes below the noise baseline. The results of this testing were:

Table 4-9. 5-D testing results, 869 a priori classes, no noise.
Epochs: 2, nodes: 228, time: 1 sec.

Change amount Δ	# changes detected	Percentage of expected detections
5%	31472	96.0%
9%	32202	98.3%

The caption to Table 4-9 shows that the programming took only one second. This was double-checked. The diversification of frequency components, through the 869 a priori classifications seemed to have provided quick programming when combined with the fewer input cases caused by ignoring the noise information. The accuracy of 96% is only slightly less than the best accuracy obtained without ignoring noise components (Table 4-8) and there are only thirteen more nodes. If learning speed is an important issue, then this technique should be used. The plot of detected changes appears in Figure 4-78. Compared with the coverage shown in Figure 4-77, it is seen that coverage is greatly improved.

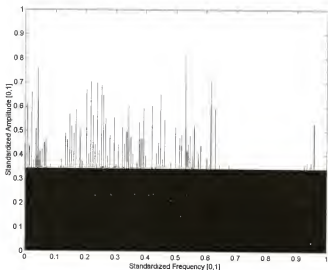


Figure 4-78. Spectrum changes detected by 5-D neural net, with 869 a priori classifications and no noise.

CHAPTER 5 CONDITION-BASED MONITORING SYSTEM

The applications of neural networks developed in this research are intended to provide modules in a turbine condition-based monitoring (CBM) system. The various vibration analysis modules can supply vital information in determining the health of the turbine. Some essential components of the neural vibration analysis modules are presented below.

Condition-based monitoring requires interpretation of the condition of machinery from sensor inputs [42]. Neural networks are being increasingly applied to this problem because of the detailed knowledge that they can contain [43], [44], [45], and [46]. Neural networks can also learn many aspects of the condition response without requiring that the response be analyzed from first principles. In many applications, it is not necessary to mathematically predict the expected response from physical phenomena. It is just required to present the neural network with examples of the response, using appropriate features, and indicate to the network the definition of the response through supervised learning techniques. The development of condition-based maintenance systems can thus be accelerated using the empirical data without lengthy analysis of the root causes of the different waveforms in the data. Of course, increased knowledge of the subject domain will improve the operation of the neural network by improving the effectiveness of the features used in the network [47] and providing more intelligent interpretation of the results.

Mode-driven knowledge set. Network weights for many operating modes must be learned by the network to provide the neural network with processing capability for many turbine operating points. As the turbine is operating, mode detection logic can select, or gate, the appropriate network weights for use in analysis of the vibration signal. A set of weights would be obtained for every portion of the possible operating range of the turbine. Interpolation between portions may be possible, given the ease of interpretation of the meanings of the Fuzzy ART classification regions, or weights. The discretization of the mode separation of the neural weight sets must be determined to provide coverage throughout most of the expected operating modes. For practical implementation, a neural network could be used to determine the operating mode, along with other analysis techniques. When the turbine is operating in a recognized mode, the system will digitize enough data for an analysis and process the input.

Neural network methods to monitor non-stationary turbine operation would rely on networks that incorporate time into their input set. A time-frequency representation of the turbine signal could be cut into different time segments, providing an ordered sequence of feature-sets, and used to obtain a sequence of events for neural analysis [48]. The recognition of an ordered sequence of events is natural for human brains [49] [50], therefore it may prove more readily accomplished automatically using a neural network implementation instead of a sequential logic computer program. The non-stationary frequency spectrum information could be obtained with Fourier transforms, but spectral smearing would occur due to the changing frequencies in the signal.

A better method to obtain the time-frequency information may be to use the Wigner-Ville Distribution, [51], [52], and [53]. The

Wigner-Ville Distribution allows representation of non-stationary signals without smearing, but generates energy spikes in the time-frequency estimate that are unrelated to physical causes. These alias signals present a problem when using the neural network to learn a representation of the spectrum. In the simplest interpretation, the network would have to learn the spectrum and the alias signal information. This may be acceptable. The alias signals are deterministic, in that they are mathematically related to the actual signal in the spectrum; they are formed between any two signals separated by frequency or time. Because of the complexity of the turbine spectrum, many aliased signal would be generated, with possible overlaps of physical signal information. Techniques to reduce aliasing [54], such as smoothing, may be applicable, but the accuracy of the amplitudes at the output of the smoothing algorithm must be investigated.

In some machinery, such as single-shafted turbines, it is possible to set acceptance envelopes, or classification regions, around known features, where the features are proportional to the shaft rotational velocity. The spectral data would then be adjusted using a technique called order normalization to always show the shaft rotational velocity at a certain frequency in the spectrum. The relative harmonics and proportional features then always appear at the same frequency in the spectral diagram, no matter what speed that the shaft is turning. This simplifies the construction of the acceptance envelopes because they can stay in fixed locations for all of the operating modes of the machine. In the LM2500 gas turbine, there are two aerodynamically coupled shafts, rotating at independent speeds, not proportionally linked. Order normalization would only be able to be performed with respect to one shaft, thus causing problems as the other

shaft changes speed. Because of the problem of two shafts, order normalization was not used. The stored knowledge can not be represented as a simple mixture of the two shaft responses, but must be stored as multiple learned signal characterizations. Instead, the mode detection neural network would provide input to an operating mode selection subsystem that would choose the correct set of weights to use for different combinations of speeds and operation of the two shafts.

The operating mode selection subsystem would be realized as a dynamic input to the neural network. As with the use of a priori feature separation, the operating mode selector would augment the input space to store the knowledge learned from different operation modes into different section of the multidimensional network weights. When the actual turbine operation matches that of a learned mode, its inputs would be compared with the stored knowledge. Because of the wide variation of possible speeds for the two turbine shafts,

One set of classification weights for the Fuzzy ART network consumed 228 nodes, each having 10-dimensions, considering the complement values. Different classifications would use different amounts of nodes, but 228 nodes was one of the higher node counts encountered for 5-D networks. Each dimension in a node would use a single 32-bit floating-point number. Thus, the total memory required for a single node classification would be $228 \text{ [nodes/classification]} * 10 \text{ [dimensions/node]} * 4 \text{ [bytes/dimension]} = 9120 \text{ [bytes / classification]}$. Assuming 20 different speeds for each of the gas generator and power turbine shafts would require a total storage space of $9120 \text{ [bytes/classification]} * (20 * 20) \text{ [classifications]} = 3648000 \text{ bytes (3.479 MB)}$ of memory. This memory requirement is well within the reach of most embedded single-board computers.

Memory. One of the primary requisites for a monitoring system is training data for the neural networks. The data assembled for this research provides a good foundation for developing the theory, but much more data needs to be gathered and incorporated to take full advantage of the new capabilities. To facilitate the accumulation of this specialized knowledge, the condition-based monitoring system should be equipped with a large non-volatile memory storage system. As new operating conditions are encountered, the data should be recorded and shared with other installations of the CBM systems to expand the knowledge base of known turbine faults. A large bank of non-volatile memory also provides extensive trending information to be captured and analyzed. This memory could be implemented with a disk-based subsystem.

There should be a large bank of RAM for holding all the neural network weights and trending information. Non-volatile memory tends to be slower than RAM, so there should be no need to access non-volatile memory to get these parameters, except at initialization.

Processing capability. The neural networks and trending information for this research were implemented in a digital computer and are processor intensive. A modern general-purpose processor would perform the tasks well. A fast digital signal processor would be preferable for many of the tasks because signal processing algorithms such as the FFT, IIR filtering, and vector operations are used extensively. Neural network architectures are highly parallel and could make good use of parallel computing capabilities using multiple processors.

Analog electronics. There should be a low-noise, multiple input analog signal conditioning system for signal acquisition. This system should be suitably isolated from the digital noise from the computer

section. Active, anti-aliasing filter sections must be inserted before the analog-to-digital converters.

System outputs. The system outputs may be transmitted to the user through a graphic display or a communications network, or across a computer backplane. The outputs consist of trend warnings and indications of detected defects in the turbine. The outputs could be used to make charts for turbine operators to review the changing trends in the turbine vibration.

Trend warnings. An expert system can be programmed to execute rules that monitor the trend information emanating from the neural trending system. As new trend changes are detected by the neural trending system, a record is stored of each new change. This change record must include the following:

1. new magnitude of the spectral component,
2. the 1X frequencies of the power turbine and gas generator,
3. the frequency of the changed component,
4. the time and date,
5. The identification, if known, of the changed component.

The identification of the changed component can be provided by the a priori classifications of the inputs. Other identification would be algorithmically generated. At each operating point of the turbine, the array of nodes representing that operating point form a neural classification vector. If a node changes, its change is recorded and can be subsequently compared to further changes in that node. The expert system can be trained to understand the existence of sidebands and check appropriate nodes, based upon logically set up sideband relations, or upon frequency differences. Ideally, a measure of the severity of a trend would be established. The trend severity could be

based upon the duration of the trend and the height reached in the trend.

The expert system could combine the vibration information with information from other subsystems to provide more information about the machinery condition [55], [56]. Automated processing of rules concerning the limits and dependencies of equipment being monitored can improve defect detection capabilities, and increase the availability of the entire system. Various levels of expert systems could act upon information from the local control systems, increasing the ability to recognize and report impending problems, and possibly providing modifications to avoid the problem.

Indication of detected defects. The mode-detection neural network would eventually be trained with recorded defect data if the system is implemented and used. The mode-detection neural network may thus be trained to provide an indication of fault conditions. The output of the mode detection system could be used as an input to the neural trending system; thus, automatically selecting the appropriate classification patterns to use depending on turbine mode. The turbine frequencies could also be input to the neural trending system using fractional-dimension 1-of-C coding further specifying the operating mode. The neural trending system would thus choose the most appropriate internal classification sets automatically, with each classification set being dimensionally shifted from adjoining sets.

Figure 5-1 shows a block diagram of the suggested CBM system

Health assessment subsystem. Measures may be derivable from the trending data that would indicate the expected remaining usable life of the turbine before service. These could be produced by calculating the rate of change of spectral components exhibiting an increasing trend. A maximum value could be used as an upper end-point in the remaining

life calculation. The shortest time until a trend value is projected to surpass the end-point threshold would then be reported as the forecast date for service. Expert information could be used to temper this warning based upon the expected severity of the incipient problem.

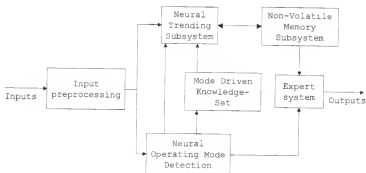


Figure 5-1. Condition-Based Monitoring System

Trending system operation. The Fuzzy ART architecture is used to learn an input vibration spectrum and to monitor subsequent spectral input for novel cases. The frequency and amplitude information provide two input dimensions to the Fuzzy ART network. A priori knowledge concerning the membership of certain features in the spectrum into known physical phenomenon provides another dimension.

The network then learns the information in the spectrum to a specified degree of vigilance during initial training. Subsequently, new cases are presented to the network and the network either recognizes the cases as being previously learned, or indicates that the new inputs do not resonate with previously learned knowledge. The fact that an unrecognized, or novel, case has occurred provides the crucial information for vibration analysis. The novel case is recorded in a

separate file along with the time of occurrence, to provide detailed trending data that can be analyzed to detect machinery degradation. The novel case can be immediately learned by the network, to allow adaptation to changing machinery conditions.

If a certain spectral component is recorded as following a trend of increasing amplitude over time, then it follows that an associated mechanical component may be suffering degradation. Maintenance activity may then be scheduled to inspect the associated component. After maintenance and repair, a set of new baselines should be obtained from the turbine for modification of the information contained in the Fuzzy ART neural network and a new monitoring history would be started. Before retraining, the original information in the network could be stored, but should then be deleted, to reduce dead nodes that no longer represent the turbine. This frees memory to provide space to create new nodes for the next period of change detection.

In the vibration monitoring application, modification of the vigilance has proven valuable. The vibration peaks, rising above a threshold, are learned and monitored with a high vigilance ($\rho^P \approx 0.95$) and the noise floor, below the threshold, is learned with a lower vigilance ($\rho^P \approx 0.15$). This technique results in fewer neurons required to perform an otherwise futile effort of attempting to learn detailed noise characteristics with high vigilance. In the LM2500 turbine application, the noise floor generally appeared at least 20 dB below major peaks, thus providing enough signal-to-noise ratio to apply this technique.

A combination of fixed-set training and on-line training would occur during the initial installation of the neural trending subsystem on a particular turbine. Before installation, the neural trending

system would be trained with the general response of the class of turbine, using a fixed-set training method. At installation, a cycle of training ensues to gather the information concerning the actual spectrum of the particular turbine for subsequent comparison in the neural trending operation. This pre-train and modify method may be useful, but could be too general in application. A better method may be to run the particular turbine at various expected operating points and perform fixed-set training with data collected directly from the turbine. This avoids problems arising from different sensor characteristics and aligns the training more closely with the turbine being monitored. This type of training will be called subject training.

Subject training plays an important part in the long-term use of the neural trending system. As the turbine is used, mechanical components will wear, causing changes in the spectrum. As usage continues, the spectrum will alter. In cases of damage, the spectrum may alter drastically. It is assumed that the neural trending system will have been recording these changes and learning the trends as the changes are occurring over time. At some point, the turbine will likely be serviced to correct problems, or as part of general maintenance. After service, the neural net should be reset to learn the new characteristics of the repaired or maintained turbine.

Online knowledge of the turbine operation, and detection of problems such as excessive resonance at certain operating speeds, can be used to prolong the operating life of turbine engines by controlled avoidance of the problematic operating modes. If excessive component vibration is detected at a certain operating speed, the control system can accelerate through that operating point to the next safe region of operation [57]. The capability to detect the excessive vibration and

to control the turbine to avoid it, requires close integration of the vibration detection system with the turbine controller. This integration would also be beneficial when the vibration detector is learning the turbine response spectrum. It may be possible to cycle the turbine through various operating states automatically while the neural network learns the responses at each operating point. The Fuzzy ART-based vibration detector, when coupled with an algorithm in the turbine controller, may thus enable emergency operation of a turbine that exhibits some excessive vibration in certain operating conditions, while minimizing vibration in the turbine.

CHAPTER 6 CONCLUSIONS

Results

A set of seven time-domain and 241 frequency-domain vibration features was developed for use as neural network input. A set of seven of these features was used to train various supervised neural networks to detect turbine operating mode. The resulting accuracy of the various networks is shown in Table 6-1.

Table 6-1. Supervised neural network classification accuracy on the vibration dataset.

Neural Network Architecture	Mode Classification Accuracy
Back-Propagation, (multilayer perceptron)	81%
Probabilistic Neural Network	83.3%
Learning Vector Quantization network	89.6%
Radial Basis Function network	76.6%
Fuzzy ARTMAP network	89.4%
Modular Neural Network	95.0%

An unsupervised network, Fuzzy ART, was used to analyze an entire 32768-point vibration spectrum. It compressed the representation of the spectrum into 228 neurons, with 10 weights each.

The inputs to the Fuzzy ART network consisted of a frequency vector, an amplitude vector, a matrix of a priori classifications and a vector of individual case vigilance values to control network precision. The a priori classification data consisted of the 241 frequency-domain features and 620 spectrum dimensional separation features, for 861 a priori features. If these inputs had been coded with 1-of-C coding, there would be 863 input dimensions, including the frequency and amplitude information. A neural network with 863 dimensions would be impractical. A technique was developed, based upon resonance field theory and the desired sensitivity of the network, to code these 861 a priori features into three analog features, using fractional-dimension 1-of-C coding.

The network sensitivity was controlled to detect changes of 5% of full scale in any spectral component. The network detected 96.3% of the changes introduced.

Contributions

Turbomachinery vibration features were researched to determine the methods of application of neural network technology to the analysis of the vibration. A set of features that allowed neural networks using supervised learning to discern the operating state of a turbine was developed. A neural network trained with a competitive learning method, Fuzzy ART, was used to detect changes in the spectrum. The monitoring of changing trends in the vibration is a standard technique in machinery health monitoring. This neural network application automates the trending process by learning the actual spectrum and responding to changes in individual spectrum components.

One of the major contributions of this research was the development of a technique to use a priori information to separate neighboring features in the input training set. This method effectively separated the neighboring features and compressed a large number of features indicators into a much smaller number of analog input dimensions was developed. For example, 819 portions of the spectrum were separated by three new network inputs. This technique required detailed knowledge of the classification properties of the network to create a technique providing robust feature separation.

The use of a priori information in the inputs to the neural network also increased the training speed dramatically. Without the a priori information, the network had to create a representation of the spectrum using spectrum components that were near to each other in the input space. The a priori feature separation allowed separation of section of the vibration spectrum into different areas of the input space, thus easing the task of the neural network. Without a priori classification, the network required 650 epochs to train, taking 2750 seconds. Using 869 a priori input separations, the network required only two epochs and less than one second to learn the same information. It reduced programming time by 99.96%, but increased the number of nodes required by 50%. Weighing the cost of memory (152 nodes, 4 dimensions each = 2.4K of memory, versus 228 nodes, 10 dimensions each = 8.9K of memory) versus the cost of programming time (45 minutes versus 1 second), it appears that a priori coding is a valuable technique.

The theory of detection of a given input change in Fuzzy ART was developed. This allowed precise determination of the learning parameters of the network, particularly the vigilance required to detect a given change.

The Fuzzy ART network operation was adjusted to accept a vigilance parameter that was adjusted for each input vector. This change reduced the number of nodes generated for a given representation of the spectrum. Input vectors below a predefined noise floor threshold were given a low vigilance, while vectors exceeding the threshold were given the high vigilance required in order to detect small changes in the spectrum. A 74% reduction in the number of nodes was seen using this method.

The theory of Fuzzy ART resonance fields was developed. This yielded a precise definition of the range of input vectors that will resonate with a given node. Resonance field analysis provided necessary information to effectively separate the input features in the input using a priori feature separation.

Conclusions

In this research, a new capability was developed for vibration analysis. The principles employed can be generalized for many applications. The neural change detector, or neural trending system, allows a computer to automatically form an internal description of the subject environment and to detect changes with adjustable sensitivity. Although it has been applied to rotating machinery vibration analysis, it may be applied to any form of spectrum analysis where it is desired to learn the characteristics of the spectrum and to detect spectral changes. In a supervised Fuzzy ARTMAP implementation, the neural network could be trained to take appropriate action based upon detected changes. The theory of resonance fields can be used in Fuzzy ARTMAP to analyze the separation of vibration classification regions allowed to trigger learned responses.

The ability to automatically create these classification regions in the frequency spectrum provides an enhanced means to implement acceptance envelope testing of complicated spectrums. Not only are all spectral features automatically identified and tracked, but also the envelopes will automatically be made as narrowband as appropriate to represent the feature. The system sensitivity can be selected using vigilance settings. The net result is a system that is sensitive to small changes throughout the spectrum, detecting changes both in known and in unexpected features.

It became obvious during the research that using a general purpose DSP system for data acquisition results in tedious effort. It required over 3000 lines of C and assembly code, written for two different processors, along with debugging of the real-time data acquisition and transfer system. This effort may be useful if the whole system will be migrated to run solely on the DSP board, but a better solution would have been to obtain a complete off-the-shelf signal acquisition system. Conversely, it was most helpful to hand code the Fuzzy ART neural architecture because it became straightforward to add capability to the network to support individual case vigilance and novelty detection reporting. The Fuzzy ART training and testing system required 2500 lines of C code. A Fuzzy ARTMAP system used for the early research required 4700 lines of code.

The supervised networks performed surprisingly well with the limited data set that was used for training, especially the Modular Neural Network. They would likely provide valuable information if defect response data is available for training. The outputs of the supervised networks could be used as a separate input dimension to the neural trending system, for monitoring and automated trending system transitions between operating modes.

The Fuzzy ART neural network provides many important benefits to an embedded condition monitoring application. It trains quickly, can learn new information without destroying old information, and the classification performance can be controlled using case vigilance and the results of the resonance field analysis.

The analysis of the resonance fields of Fuzzy ART permits an investigation into the discernment capabilities of the network. An understanding of the sizes and orientation of the resonance fields generated by classification regions allows effective controls to be used to separate closely spaced classification regions. Without knowledge of the resonance fields, the results obtained from the neural trending system would have been poor. Overlapping resonance fields would erroneously classify changes, due to the close proximity of frequency components in the standardized spectrum. With knowledge of the maximum extents of the resonance fields, a fractional-dimension 1-of-C coded input matrix was developed to effectively dimensionally separate adjacent frequency components. Although the resonance field is hyperdimensional, knowledge of its maximum extents allowed precise determination of the amount of dimensionality to add in order to force separation of the features, permitting correct classification.

In some neural networks, the separation of features is important to enhance classification. In this application, the separation of features, using fractional-dimension 1-of-C coding, was used to ensure correct nonclassifiability of features. Only in this way could changes in the features be reliably detected. The results of the resonance fields analysis permitted control of the maximum change permitted before a component would no longer resonate with the node that previously classified it. The fractional-dimension data sets were

generated dynamically based upon turbine rotational velocities and used as input to the network for training and testing.

Individual case-vigilance techniques were also used to minimize the size of the compressed internal representation of the spectrum and to remove most of the spectral noise. After leveling of the noise floor, the vigilance levels of individual frequency components were dynamically inserted to a case vigilance vector. The case vigilance vector was presented to the network during training and testing to control the precision of classification. Higher vigilance was used for spectral components above a noise floor threshold.

A condition-based maintenance tool was proposed, incorporating the Fuzzy ART neural trending system and a supervised network for mode detection. The system could be manufactured as a stand-alone computer installation, providing continuous monitoring of changing trends in the vibration of major rotating elements. The training regimen for this system would consist of supervised training of the mode detection system, followed by unsupervised training of the Fuzzy ART system. Outputs from the trained mode detection system would be used to dimensionally shift input knowledge to different parts of the Fuzzy ART long term memory.

Neural networks will likely provide many benefits to vibration analysis, as well as many other problem domains. Their use increases with value with the level of knowledge used in their training and implementation. One of the most important requirements for successful operation is a well-developed training dataset. The time and effort spent developing the training data should be at least commensurate with the effort put forth to develop the neurocomputer system.

Recommendations for Further Research

Neural network technology encompasses some of the best developments in mathematics, statistics, system theory, digital signal processing, neurobiological research, and computer engineering to provide new capabilities for the creation of machines incorporating computational intelligence. It is obvious that the application of neural network technology will have a dramatic impact on machinery of the future. New capabilities, such as discussed in this research, provide machines with more and more of the properties conventionally considered to be exclusively human. As in any technological achievement, the creation of a working neural network product will likely be difficult. A neural network represents only a small portion of a total system designed to handle a problem. Data must be collected, preprocessed, and transmitted to the neural network for analysis. The results of the analysis must be interpreted and need to cause resulting actions. The network must be trained and updated as required to adapt to a changing environment.

The creation of a neural network system and its training regimen requires not only knowledge of the computational and architectural properties of the network and its embedded environment, but also a knowledge of the application domain. The research included an analysis of the turbine operation, with mechanical interpretation of the vibration components. Knowledge of the application may be a hindrance to the widespread use of neural networks. Not only do the engineers have to be cognizant of neural network technology, but must also have an in depth knowledge of the problem to be solved. A multidisciplinary team, including subject-matter experts having knowledge of the

application domain, will likely provide the most efficient method to implement a neural network application.

The architecture of the Modular Neural Network, with local experts, seems to provide a direction for future research, in that the outputs of specialized networks are combined to synthesize a final output. Fuzzy ART also allows virtual network modularization through dimensional shifting. The use of groups of neural networks, trained in separate tasks, may provide an important step forward in the application of computational intelligence.

Computational intelligence architectures that have rich enough training and enough auxiliary system resources to operate and learn about their application area will eventually be developed and standardized. Small working applications will be combined into larger ones and the technology may expand beyond the capabilities of programmers and analysts to understand. But it seems that it will forever be impossible to fully duplicate the operation of even the smallest brain, that this capability will always be out of reach. Even so, the computing capability provided using neurobiological models greatly increases the usefulness of machine intelligence.

Some directions for future research include

1. Investigate modifications to the Fuzzy ART architecture to allow resonance field size to be altered from dimension to dimension. This modification would reduce the need for dimensional-shifting to separate closely spaced features.
2. Enhance the network architecture to allow monitoring of non-stationary turbine effects. This may include recording the entire spinup or coastdown of the turbine and analyzing it by sections using a Wagner-Ville Distribution. Transient noise

patterns can also be detected and analyzed to detect potentially damaging impacts [58].

3. Use temporal neural networks to monitor the vibration waveform for bearing impact noise or other time-domain transients. The roller and ball bearings on the gas turbine tend to be mounted interior to the rotor, increasing the transmission path for bearing vibration to reach the turbine case. It may be possible to detect scraping blades using temporal networks. The actual defect response waveforms would need to be obtained for use in training.
4. Investigate the use of temporal neural networks to analyze the trends detected in the turbine data. While the trends occur over a time-base of weeks or months, digital implementations of the temporal networks could maintain their state over these periods. The trends of the changes would be trained into the networks, helping to determine maintenance periods for equipment, and predicting remaining useful life of the equipment.
5. Develop an expert system to respond to trends detected by the neural trending system. This system should have the rules coded that permit it to associate sidebands with main features, to apply trending limits, to monitor for speed of trend changes, and to adjust to unexpected trending features. This system should drive the operator or network interface. Its outputs should provide conclusions of general turbine health, as well as the expected duration until trend limits are exceeded, and indicate which components are showing the most significant wear or excessive vibration, thus enhancing maintenance effectiveness.

APPENDIX A DIGITIZING EQUIPMENT

A system was constructed to digitize taped turbine signals for this research. Two implementations of the system were constructed: a high-speed sampler and a low-speed sampler. Memory availability and real-time constraints in the data transfer to the personal computer (PC) from the digitizer required two different software implementations. The low-speed sampler was also required to implement a pair of 35th-order, digital, infinite impulse response filters to increase the capabilities of the input anti-aliasing filter. A TMS320C31 Digital Signal Processor (DSP) board was coupled to a personal computer (PC) using semaphore logic implemented in a dual-ported RAM (DPRAM) system that allowed communication across the ISA bus of the PC. The DPRAM was used to transfer the resulting vibration and tachometer data to the PC using ping-pong memory transfers, in which the PC would read one section of the DPRAM while the DSP wrote to another section. When the signals were obtained, the C code written for the PC converted the signals into Matlab matrix structures and stored the information on disk.

The real-time software written for the DSP consisted of C and assembly code, built using the Texas Instruments C compiler and assembler. The code controlled a 16-bit digitizer board and a 12-bit digitizer daughterboard. Interrupt logic for sampling and semaphore logic for PC communications were implemented, along with real-time infinite impulse response (IIR) filters. The PC software consisted of

C code to initialize the DSP card, communicate with the DSP card for setup and data transfers, final signal processing, and creation of Matlab matrix files for hard drive storage of the digital signals.

A block diagram of the digitizer is shown in Figure A-1. An illustration of the digitizer system is presented in Figure A-2, where the system was in use at the Florida Power Cogeneration Plant located at the University of Florida campus to record their LM6000 gas turbine.

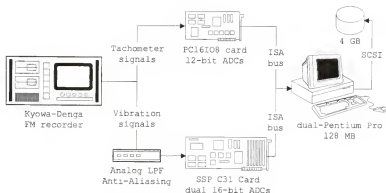


Figure A-1. Digitizing equipment.

High-Frequency Sampler

The high-speed sampler performed two channels of 16-bit digitization at 40026.06 [samples/second] for the accelerometer signals and two channels of 12-bit digitization at 20013.03 [samples/second] for the tachometers. The system recorded 65536 samples of each vibration signal and 32768 samples of each tachometer signal at each activation. This was limited by the amount of DSP card on-board RAM. The high-speed digitizer transferred its data to the PC when the

sampling was complete, as opposed to the low frequency sampler, which continuously streamed data to the PC during the sampling.

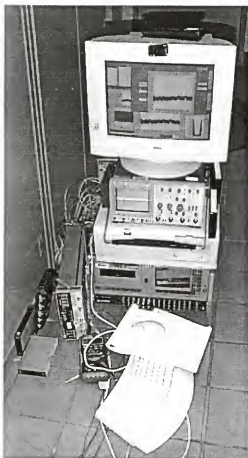


Figure A-2. Digitizer and data recorder system.

Low-Frequency Long-Duration Sampler

This system was used to digitize the entire contents of a recorded turbine vibration tape. The results of this sampling are shown in the spectrograms of Chapter 2. The low-speed data sets were originally going to be used for analysis of the turbine once-per-rev frequencies and their harmonics. The high-speed data contained the same information and more. The Fuzzy ART system proved capable of handling the larger amount of spectral data that resulted. The low-speed data still proved useful in discernment of turbine operating modes for generation of the training data for the supervised neural networks.

The sampler that was developed to obtain a fine-grained spectrum through 20 kHz used too much on-board DSP memory to obtain a lengthy signal. The LM2500 signals were recorded on the FM recorder at a 38cm/sec rate, storing a maximum of 10 minutes per tape. If the entire 10 minutes were to be recorded at 40ksamples/second for four channels and converted to PC double-length floating point data (8 bytes long), the resulting disk files would have been approximately 732 MB long. The signals would not have been able to be stored in the 128-MB PC memory and would have necessitated continuous buffering of intermediate data to disk. While this sampling rate may be appropriate for on-line monitoring where ongoing data can be processed and discarded, for off-line data collection the data size was too immense for storage using the system at hand. To perform the analysis, the high-speed data sets were recorded at separated locations in the input tape, while the low-speed data sets were recorded at length.

In a prior implementation of a DSP vibration analyzer, dedicated hardware was used for tachometer conditioning and frequency detection

employing 6 MHz clocks and 32-bit counters integrated with the interrupt capabilities of the DSP. In the neural system, the tachometer implementation was constrained by the capabilities of the commercial off the shelf equipment being used. Careful analysis was required to create a system that would provide less than 1 Hz error for a tachometer frequency of less than 9400 Hz.

The tachometer signals had a higher bandwidth requirement than the vibration signals. The maximum tachometer signal bandwidth had to encompass the highest turbine speed multiplied by the gear ratio. The tachometer signal consisted of a single sine wave having a frequency that matches the turbine speed multiplied by the gear ratio of the gear that is monitored by the magnetic pickup sensor. This turned out to be the greater of $70 \text{ Hz} \times 83 = 5810 \text{ Hz}$ for the power turbine or $200 \text{ Hz} \times 47 = 9400 \text{ Hz}$ for the gas generator. The tachometer rate was chosen as 20013 samples/sec to ensure that the maximum tachometer signal frequency would remain below the Nyquist rate (10,007 Hz in this case). This frequency was also an even multiple of the sampling rate of the vibration signals, allowing a group of tachometer readings to be obtained and passed to the PC in synchrony with the vibration signals.

The tachometer signals were recorded as sine waves. The only information that is needed to determine the frequency is the times of zero-crossing. A combined DSP/PC technique was created to just transmit zero-crossing information to the PC and convert that to tachometer frequency in real-time. The zero-crossing information required just one bit for representation: 1 for positive, 0 for negative. These bits were packed into 16-bit words for transmission to the PC. The PC would monitor the tachometer input for a change from negative to positive since the last sample, indicating that zero has been crossed.

In order to obtain accurate zero crossing information, an adaptive technique was developed to allow the DSP to find the mean of the tachometer signal before transmitting information. The mean had to be found because the tachometer was riding on a DC level that was not always constant. The tachometer sensing could be termed an adaptive mean-crossing detector.

APPENDIX B ANTI-ALIASING FILTER

In order to perform low-speed sampling for the tape survey, a flat, accurate and high-order filter was required to prevent frequencies higher than the Nyquist rate from corrupting the sampled data with aliasing effects. The cutoff frequency for this low-pass filter was chosen as 600 Hz, to capture the first two harmonics of both the power turbine and the gas generator rotational velocities.

Careful design of the anti-aliasing filter system was required because a large amount of combustion noise was present in the spectrum immediately above the cutoff frequency of the filter. The noise had a nearly Gaussian distribution with a tail that began at approximately 815 Hz with a peak at approximately 850 Hz. This complicated the filter requirements. To maintain 1 dB accuracy within the passband of 0 to 600 Hz, while dropping 60 dB in the stopband 200 Hz away, required a hybrid analog/digital filter. The sampler still required an analog filter to physically reduce the higher frequency signal amplitude before digitization. Two filter channels were required, one for each of the vibration signals.

Butterworth filters were created for both the analog and digital sections to achieve minimum ripple in the passband. The input signal was 4x oversampled to allow the digital filter to have frequency space to perform the needed attenuation without becoming too large for efficient implementation or unwanted group delay response. The filtered signal was decimated to get the proper sampling rate after

filtering. The oversampling also allowed a lower order analog filter to be created, minimizing dependency on component tolerances.

The filters were implemented with a net cutoff frequency of $F_c = 625.4$ Hz, a sampling rate of 5003.258 Hz (for oversampling), and an analog filter stopband of 4377.8 Hz. The analog filter cutoff frequency was calculated to be 778 Hz, to support the 1 dB passband requirement. The resulting sampling rate of 1251 samples/sec was transmitted to the PC.

Aliasing effects associated with the sampling frequency were taken into account to allow controlled aliasing to occur. The analog filter stopband ensured that, although aliasing was allowed to occur, the effects would not be present in the final 0-600 Hz passband. This technique permitted the sampling frequency to be halved and decreased the load on the DSP.

Analog Filter Section

The analog section consisted of a 4th order Butterworth active filter, implemented with FET-input op-amps. The SPICE analysis circuit is shown in Figure B-1. Two of these circuits were built, using precision resistors and handpicked capacitors. An input buffer op-amp and an output buffer op-amp were added, along with +/- 15V power supply decoupling capacitors. The resulting circuitry was packaged in a shielded enclosure, as shown in Figure B-2.

The passband, through 600 Hz, had a maximum loss of -0.81639 dB, within the design specs of -1 dB, and even less ripple. At the beginning of the analog stopband (4378 Hz), the analog filter response was -66.5334 dB and decreasing from there.

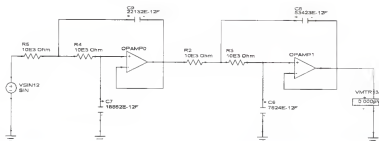


Figure B-1. Anti-aliasing filter SPICE circuit.

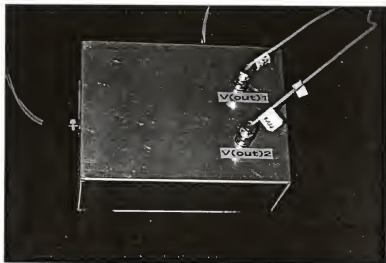


Figure B-2. Anti-aliasing filter implementation.

Computer-aided trade-offs were conducted to find an acceptable order for the analog filter that would provide the needed stopband attenuation, while minimizing the passband attenuation.

Digital Filter Section

Dual 35th-order digital Infinite Impulse Response (IIR) filters were coded in TMS320C31 DSP assembly language and implemented in real-time to process the vibration signals after they were analog-filtered and digitized. Each filter was realized as a cascade of 18 biquad (second-order) sections of the form shown in Figure B-3. The a and b multiplicands define the operating characteristics of the filter [59], [60]. The z^{-1} notation indicates unit time-sample delays. The $d(n)$ nodes indicate internal filter states.

The second-order system multiplicands were determined using the Matlab signal processing toolkit. Filter types and group delay activity were analyzed using the Monarch DSP analysis and design system from The Athena Group.

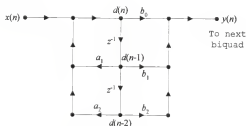


Figure B-3. DSP IIR biquad signal flow graph.

The performance of the anti-aliasing filter can be seen in the spectrograms of Chapter 2, where no aliased components of the combustion noise are visible.

LIST OF REFERENCES

1. Traver, D. and Foley, D., "Diagnosing gear and motor problems with signal analysis," in *Mechanical Signatures Analysis: Machinery Vibration, Flow Induced Vibration and Acoustic Noise*. American Soc. of Mechanical Engineers, New York, 1987.
2. Atherton, W.J. et al, "Automated acoustic intensity measurements and the effect of gear tooth profile on noise," in *Mechanical Signatures Analysis: Machinery Vibration, Flow Induced Vibration and Acoustic Noise*. Amer. Soc. of Mechanical Engineers, New York, 1987.
3. Mathew, J. and Alfredson, R.J., "The monitoring of gearbox vibration operating under steady state condition," in *Mechanical Signatures Analysis: Machinery Vibration, Flow Induced Vibration and Acoustic Noise*. Amer. Soc. of Mechanical Engineers, New York, 1987.
4. Roemer, M.J, "Real-time health monitoring and diagnostics for gas turbine engines," in *Proceedings of International Gas Turbine & Aeroengine Congress and Exhibition*. Amer. Soc. of Mechanical Engineers, New York, 1997.
5. General Electric Co., *Technical Manual for LM2500 Propulsion Gas Turbine, Description, Operation and Installation*. Marine and Industrial Engines and Service Division, General Electric Co., Cincinnati, OH, 1987.
6. Broch, J.T., *Mechanical vibration and shock measurements*. Bruel & Kjaer Instruments, Inc., Naerum, Denmark, 1984.
7. Mitchell, J.S., *Introduction to Machinery Analysis and Monitoring*. PennWell, Tulsa, OK, 1993.
8. Robinson, J.C, Ryback, J.M., and Saller, E.R, "Using accelerometers to monitor complex machinery vibration," *Sensors*, June 1997.
9. Nadler, M and Smith, E.P., *Pattern Recognition Engineering*. John Wiley & Sons, New York, 1993.
10. Jones, B., *Matlab Statistics Toolbox Users Guide*. The MathWorks, Inc., Natick, MA, 1996.
11. Peebles, P.Z., *Probability, Random Variables, and Random Signal Principles*. McGraw-Hill, New York, 1987.
12. White, G., *Machinery Vibration*. DLI Engineering Corp., Bainbridge Island, WA, 1995.
13. Rao, P., Taylor, F.J., and Harrison, G.A., "Real-time monitoring of vibration using the Wigner distribution," *Sound and Vibration*, pp. 22-25, May 1990.

14. Ehrlich, F., "The influence of trapped fluids on high speed rotor vibration," *Proc. Amer. Soc. of Mechanical Engineers Vibrations Conference*, Boston, 1967.
15. Day, I.J et al, "Stall inception and the prospects for active control in four high speed compressors," in *Proceedings of International Gas Turbine & Aeroengine Congress and Exhibition*. Amer. Soc. of Mechanical Engineers, New York, 1997.
16. Minsky, M. and Papert, S., "Perceptron," excerpt in *Neurocomputing: foundations for research*, MIT Press, Cambridge, MA, 1988.
17. Haykin, S., *Neural Networks, A Comprehensive Foundation*. Macmillan, New York, 1994.
18. Freeman, W.J., "Linear analysis of the dynamics of neural masses," in *Neuro-control systems theory and applications*. IEEE Press, New York, 1994.
19. Hopfield, J.J. and Tank, D.W., "Computing with neural circuits," in *Neuro-control systems theory and applications*. IEEE Press, New York, 1994.
20. de Vries, B. and Principe, J.C., "The gamma model—a new neural model for temporal processing," *Neural Networks*, vol. 5 pp. 565-576, 1976.
21. Catfolis, T. "Mapping a complex temporal problem into a combination of static and dynamic neural networks," in *Time in Neural Networks*, SIGART Bulletin, Vol. 5 No. 3, pp. 23-28, 1994.
22. Villarreal, J.A. and Shelton, R.O., "A space-time neural network for processing both spacial and temporal data," microfiche, NASA case no. MSC-21874-1, 1991.
23. Reiss, W. and Taylor, J.G, "Storing temporal sequences," *Neural Networks*, Vol. 4, pp. 773-787, 1991.
24. Hecht-Nielsen, R., *Neurocomputing*. Addison-Wesley, Reading, MA, 1990.
25. NeuralWare, Inc., *Neural Computing—A Technology Handbook for Professional II/Plus and NeuralWorks Explorer*. NeuralWare, Inc. Pittsburgh, PA, 1995.
26. Carpenter, G. A. and Grossberg, S., "A massively parallel architecture for a self-organizing neural pattern recognition machine" in *Neural Networks and Natural Intelligence*, MIT Press, Cambridge, MA, 1988.
27. Grossberg, S. "Recent developments in the design of real-time non-linear neural network architecture," in *Proceedings of IEEE First International Conference on Neural Networks*. San Diego, CA, pp. 1-77, 1987.
28. Kosko, B., *Neural Networks and Fuzzy Systems—A Dynamical Systems Approach to Machine Intelligence*. Prentice Hall, Englewood Cliffs, NJ, 1992.
29. Zadeh, L.A., "Fuzzy sets," in *Neuro-Control Systems Theory and Application*. IEEE Press, New York, 1994.
30. Demuth, H. and Beale, M., *Neural Network Toolbox User's Guide*. The MathWorks, Inc., Natick, MA, 1996.

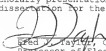
31. Frean, M., "The Upstart Algorithm: a method for constructing and training feedforward neural networks," *Jour. Neural Computation*, Vol. 2, pp. 198-209, 1990.
32. Carpenter, G.A., Grossberg, S., and Rosen, D.B., "Fuzzy ART: fast stable learning and categorization of analog patterns by an adaptive resonance system," *Neural Networks*, Vol. 4, pp. 759-771, 1991.
33. Carpenter, G.A. and Grossberg, S., "Learning, categorization, rule formation, and prediction by fuzzy neural networks," in *Fuzzy Logic and Neural Network Handbook*, pp. 1.3-1.45, McGraw-Hill, New York, 1996.
34. Huang, J., Georgiopoulos, M., and Heileman, G.L., "Fuzzy ART properties," *Neural Networks*, vol. 8, No. 2, pp. 203-213, 1995.
35. Jenkins, G.M. and Watts, D.G., *Spectral Analysis and its Applications*. Holden-Day, San Francisco, CA, 1968.
36. The MathWorks, Inc., *Signal Processing Toolbox User's Guide*. The MathWorks, Inc., Natick, MA, 1996.
37. Oppenheim, A.V. and Schaffer, R.W., "Discrete-Time Signal Processing," Prentice Hall, Englewood Cliffs, NJ, 1989.
38. Harris, F.J., "On the use of windows for harmonic analysis with the DFT," *Proc. IEEE*, vol. 66, pp. 51-83, 1978.
39. Blackman, R.B. and Tukey, J.W., *The Measurement of Power Spectra from the Point of View of Communication Engineering*. Dover, New York, 1959.
40. Guttman, I., Wilks, S.S., and Hunter, J.S., *Introductory Engineering Statistics*. John Wiley & Sons, New York, 1971.
41. Lawrence, S. et al, "On the distribution of performance from multiple neural-network trials," *IEEE Trans. Neural Networks*, Vol. 8, pp. 1507-1517, Nov. 1997.
42. Brownell, T. A., "Neural Networks for Sensor Management and Diagnostics," in *Proc. IEEE National Aerospace and Electronics Conference*, NAECON, vol. 3, 1992.
43. Kim, J.H. and Kissel, R., "Vehicle health management using adaptive techniques - CDDF final report No. 92-12", NASA Technical Memorandum, NASA-TM-108475, N95-15609, microfiche, 45 pp., 1994.
44. Jiang, H. and Penman, J., "Using Kohonen feature maps to monitor the condition of synchronous generators," in *Proceedings of the Workshop on Neural Net Applications and Tools*, IEEE Press, New York, 1993.
45. Guillot, M. et al, "Process monitoring and control," in *Artificial Neural Networks for Intelligent Manufacturing*. Chapman & Hall, New York, pp. 371-397, 1994.
46. Kanelopoulos, K., Stamatis, A., and Mathioudakis, K., "Incorporating neural networks into gas turbine performance diagnostics," in *Proceedings of International Gas Turbine & Aeroengine Congress and Exhibition*. Amer. Soc. of Mechanical Engineers, New York, 1997.

47. Duda, R.O. and Hart, P.E., *Pattern Classification and Scene Analysis*. John Wiley & Sons, New York, 1973.
48. Malkoff, D. B., "Detection and Classification by Neural Networks and Time-Frequency Distributions," in *Time-Frequency Signal Analysis, Methods and Applications*. Wiley, pp. 324-348, Hong Kong,
49. Aristotle, "De memoria et reminiscentia". ca 400 BC, Aristotle on Memory, Translator: Richard Sorabji. Reprinted in *Neurocomputing 2, Directions for Research*. MIT Press, Cambridge, MA, pp. 1-10, 1990.
50. Weiner, N. "Cybernetics, Intro to First Edition," in *Neurocomputing, Directions for Research*. MIT Press, Cambridge, MA, pp. 11-41, 1988.
51. Boashash, B. and Black, P.J., "An efficient real-time implementation of the Wigner-ville Distribution," *IEEE Trans. Acoust. Speech, Signal Processing*, Vol. ASSP-35, no. 11, Nov., 1987.
52. Classen, T.A.C.M. and Mecklenbrauker, W.F.G., "The Wigner distribution-A tool for time-frequency signal analysis. Parts 1, 2 and 3," *Philips J. Res.*, Vol. 35, pp. 217-389, 1980.
53. Taylor, F.J. and Chester, D., "The Wigner Distribution in speech processing applications," *J. Franklin Inst.*, vol. 318, no. 6, pp. 415-430, 1984.
54. Jeong, J. and Williams, W.J., "Alias-free generalized discrete-time time-frequency distributions," *IEEE Trans. Sig. Proc.*, Vol. 40, No. 11, Nov. 1992.
55. Ponnar, V., Chouhan, A.S., and Rangachari, P.J., "An integrated knowledge based approach for on-line condition monitoring & fault diagnostics of marine propulsion machinery." in the *Proceedings of the Eleventh Ship Control Systems Symposium*, U. Southampton, Southampton, England, 1997.
56. Harrison, G.A., "Neural networks and genetic algorithms in machinery control systems," in the *Proceedings of the Eleventh Ship Control Systems Symposium*, U. Southampton, Southampton, England, 1997.
57. Millsaps, K.T. and Reed, G.L., "Reducing lateral vibrations of a rotor passing through critical speeds by acceleration scheduling," in *Proceedings of International Gas Turbine & Aeroengine Congress and Exhibition*. Amer. Soc. of Mechanical Engineers, New York, 1997.
58. Maurer, W.J., Dowla, F.U., and Jarpe, S.P., "Seismic event interpretation using self-ordered Neural networks," in *Applications of Advanced Neural Networks III*, SPIE Proceedings, Vol. 1709, Bellingham, WA, 1992.
59. Taylor, F.J., *Digital Filter Design Handbook*. Marcel Dekker, Inc., New York, 1983.
60. Strum, R.D. and Kirk, D.E, *First Principles of Discrete Systems and Digital Signal Processing*. Addison-Wesley, Reading, MA, 1988.

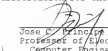
BIOGRAPHICAL SKETCH

Gregory Anthony Harrison performs design and engineering at Lockheed Martin Corporation. He has been with the company for 12 years. His current project is Internet communication systems for the X-33 next-generation space shuttle. Previously he has led Marine Control's Internal Research and Development efforts and he designed and productized a digital signal processing system for turbine vibration analysis based upon the Wigner-Ville Distribution. He graduated from the University of Florida in 1989 with a Master of Science degree in electrical engineering.

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Fred V. Layman, Chairman
Professor of Electrical and
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Jose C. Principe
Professor of Electrical and
Computer Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Antonio A. Arroyo
Associate Professor of
Electrical and Computer
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

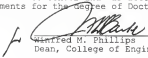

John G. Harris
Assistant Professor of
Electrical and Computer
Engineering

I certify that I have read this study and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.


Elmer C. Hansen
Assistant Professor of
Mechanical Engineering

This dissertation was submitted to the Graduate Faculty of the College of Engineering and to the Graduate School and was accepted as partial fulfillment of the requirements for the degree of Doctor of Philosophy.

December, 1997


Winifred M. Phillips
Dean, College of Engineering


Karen A. Holbrook
Dean, Graduate School

LD
1780
199_
.H319

UNIVERSITY OF FLORIDA



3 1262 08554 9052